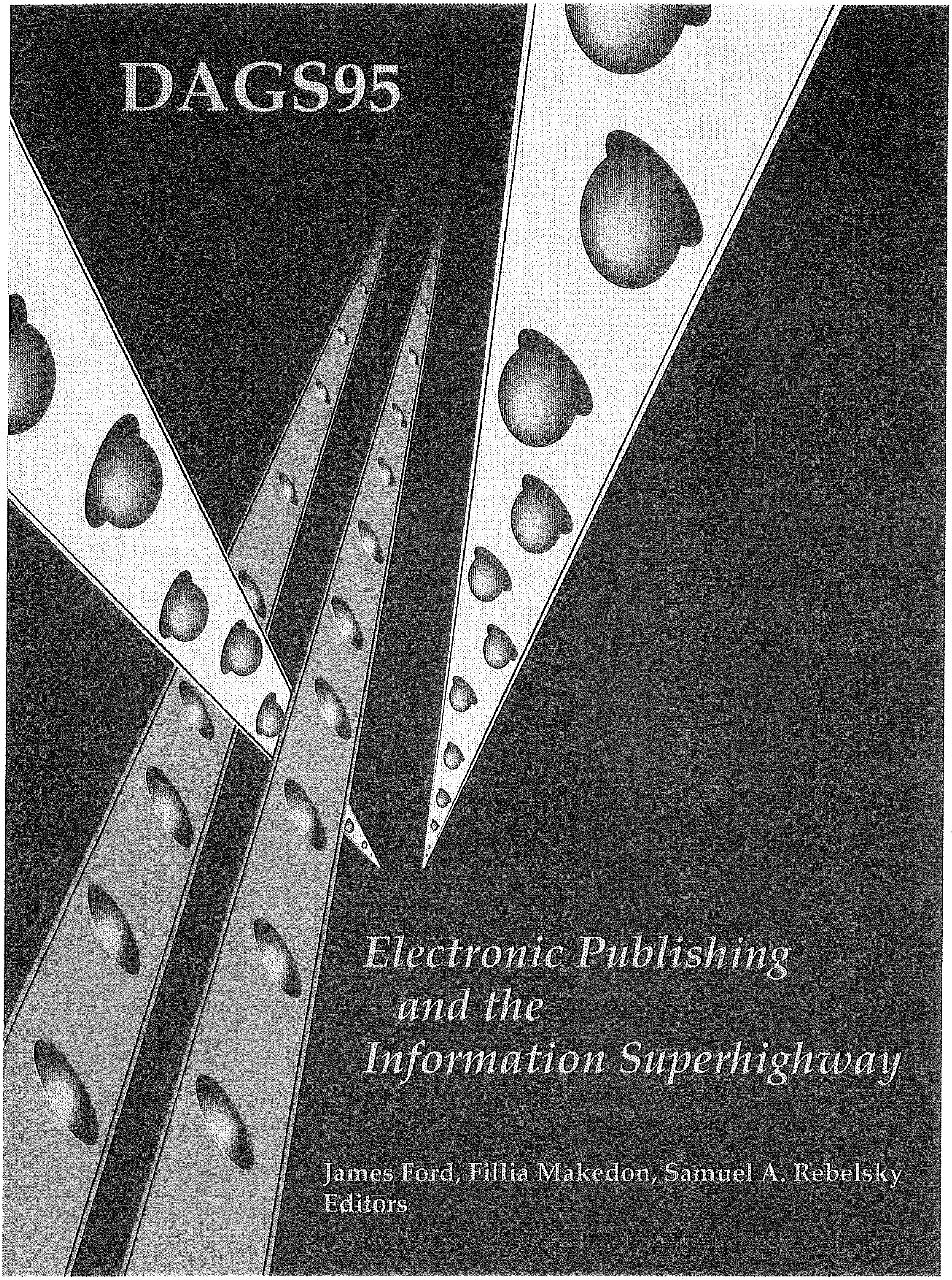


DAGS95

Electronic Publishing and the Information Superhighway

James Ford, Fillia Makedon, Samuel A. Rebelsky
Editors

About the Cover: Proceedings cover produced by Charles Owen and Samuel Rebelsky.
DAGS logo designed by Ark Lemal.

DAGS

Electronic Publishing and the Information Superhighway

Enabling Technologies • Issues • Applications

May 30–June 2, 1995 Boston, Massachusetts

Proceedings of the Dartmouth Institute for Advanced Graduate Studies

James Ford
Fillia Makedon
Samuel Rebelsky
Editors

Birkhäuser
Boston • Basel • Berlin

James Ford
Dartmouth Institute for Advance Graduate Studies
Dartmouth College
Hanover, NH 03755

Fillia Makedon
Dartmouth Institute for Advance Graduate Studies
Dartmouth College
Hanover, NH 03755

Samuel Rebelsky
Dartmouth Institute for Advance Graduate Studies
Dartmouth College
Hanover, NH 03755

ISBN 0-8176-3846-6

Printed in USA

SPONSORS

Addison-Wesley Interactive
AI Systems
Bolt Beranek and Newman Inc. (BBN)
Birkhäuser Publishers
Bremer Associates
Dartmouth College
Dartmouth Experimental Visualization Laboratory (DEVLAB)
Kiewit Computation Center
PWS Publishers
Renaissance Digital
Union Bank of Switzerland

CHAIRS

Program Chair: Fillia Makedon (Dartmouth College)
Program co-Chair: Samuel Rebelsky (Dartmouth College)
Panels Chair. Donald Kreider (Dartmouth College)
Posters Chair: John Buford (U. Mass-Lowell)
Tutorials and Workshops Chair: Panagiotis Metaxas (Wellesley College)
Audio/Visual Chair: Charles Owen (Dartmouth College)
Volunteers Chair: Michael Ferrantino (Boston University)

STEERING COMMITTEE

Fillia Makedon (chair), Scot Drysdale, Lawrence Levine, Panagiotis Metaxas, Samuel Rebelsky

PROGRAM COMMITTEE

Fillia Makedon(Dartmouth/CS), *Chair.* Samuel A. Rebelsky (Dartmouth/CS), *Co-Chair.*
Bob Allen (Bellcore); Jon Appleton (Dartmouth/Music); John Buford (U. Mass-Lowell); Jon Crowcroft (U. College London-England); Steve Cunningham (CSUS, SIGGRAPH); George Cybenko (Dartmouth/Engineering); John Danskin (Dartmouth/CS); Chip Elliott (BB&N); Domenico Ferrari (U.C. Berkeley); Otmar K.E. Foelsche (Dartmouth/Language Resource Center); Ed Fox (Virginia Tech); Peter Gloor (UBS-Switzerland); Michael Goodrich (Johns Hopkins); Carey Heckman (Stanford Law); Joseph Henderson (Dartmouth/Medicine); Albert Henning (Dartmouth/Engineering); David Karger (Bell Labs; MIT); Tom Leighton (MIT); Thomas Little (Boston U.); Hermann Maurer (Graz U. of Tech.-Austria); P. Takis Metaxas (Wellesley); Michael O'Donnell (U. Chicago); Andrew Odlyzko (Bell Labs); Maria C. Pantelia (U.N.H.); Grammati Pantziou (U. Central Florida); Paolo Paolini (Milano-Italy); Ian Parberry (U. North Texas); Steven Pemberton (CWI-Amsterdam); Larry Polansky (Dartmouth/Music); Daniel Richards (Dartmouth/Medical Libraries); Isidore Rigoutsos (IBM); Daniela Rus (Dartmouth/CS); David Sherman (U. Bordeaux-France); Janos Simon (U. Chicago); Randall Stewart (U. Utah, Hermes Pub.); James Storer (Brandeis); David Tennenhouse (MIT); Costantino Thanos (CNR-Italy); Chris Welty (Vassar); Mark J. Williams (Dartmouth/Film)

ADVISORY BOARD

Bob Allen (Bellcore); Jane Bassick (Dartmouth-Hitchcock Visual Media); Bryce Bastian (Olympus); James Breeden (Tucker Foundation); Terry Ehling (MIT Press); Charles Fenton (Renaissance Digital Publishing); Borko Furht (Florida Atlantic U., J. MM Tools and Appl.); Jay Heinrichs (Dartmouth/Alumni Magazine); Peter Hirshberg (Fusion Group); Bruce Judson (Time Inc.); Paddy Kalesh (Eyesaver); Donald Kreider (MAA, Dartmouth/Math); Jonathan Newcomb (Simon & Schuster); Peter Prichard (USA Today); Robert Prior (MIT Press); Barbara Simons (IBM); William Stahl (AI Systems); Michael Sugarman (PWS Publishers); Tay Vaughan (Timestream); Jeffrey Weitzman (Lexis Counsel Connect); Allan Wylde (TELOS/Springer-Verlag); Wayne Yuhasz (Birkhäuser Publishers)

PROCEEDINGS EDITORS

James Ford, Fillia Makedon, and Samuel Rebelsky (Dartmouth College)

ELECTRONIC PROCEEDINGS SENIOR EDITORS

Peter Gloor (Union Bank of Switzerland); Fillia Makedon and Samuel Rebelsky (Dartmouth College)

ELECTRONIC PROCEEDINGS ASSISTANT EDITORS

James Ford, Charles Owen, and Qin Zhang (Dartmouth College); Mark Moline (Addison-Wesley Interactive), Oliver Van Ligten (Union Bank of Switzerland)

Table of Contents

PANELS **293**

Preface

Many communities—including academicians, librarians, publishers, and technologists—emphasize the creation, organization, and dissemination of information. These fields are converging in the realm of electronic publishing, in which documents are created, organized, annotated, edited, distributed, and read digitally. Electronic publishing is an exciting, new, and rapidly changing field that promises many benefits. Electronic documents may be distributed to a broader audience at a lower cost. Electronic publications can incorporate nontextual materials, including interactive components and time-based media such as audio and video. Electronic publishing also provides greater facilities for organizing, retrieving, and presenting information.

Each community brings a different perspective and set of experiences to electronic publishing. Unfortunately, discussions of electronic publishing stay within each community, which often means that one community does not know about or understand the findings of the other communities. To bring these different groups together and thereby help remedy this problem, the Dartmouth Institute for Advanced Graduate Studies (DAGS) has organized DAGS95: Electronic Publishing and the Information Superhighway. The participants in the conference present the perspectives of computer scientists, human factors specialists, librarians, lawyers, social commentators, authors, publishers, information owners and providers, and readers. The papers and panel summaries in this proceedings describe issues from these many perspectives, including the design, use, and future of the World Wide Web; the roles of nontextual media in electronic publishing; the social impact of electronic publishing; effects of electronic publishing on education and the academic communities; organization and retrieval methods; and historical lessons about the development of other new media.

The World Wide Web (WWW) is perhaps the fastest growing electronic publishing system, and has introduced many to the wonders and possibilities of electronic publishing. In his article, Tim Berners-Lee, the inventor of the WWW, discusses the future of the WWW and the consortium of institutions that support the web. Michael Palmer, Hope Greenberg, Robert Duffy, and Jennifer Yacovissi provide more specific perspectives on the WWW and on different types of electronic publishing that the WWW permits, from business publishing to counter-culture "zines." Harvey Plantinga describes past experiences providing large information repositories on the WWW and Stephan Spencer, Jean-Yves Sgro, and Max L. Nibert suggest a novel publishing application for the WWW: publishing virus structures in multimedia formats. The popularity of the WWW also creates problems: there is often so much out there that one gets "lost in the hyperspace." Yee-Hsiang Change and Ellis Chi describe a way of viewing and organizing the information on the WWW that provides better grounding for the reader.

Electronic publications that incorporate other media, such as video and text, must be more than "electronic videotapes," they must also provide ways to analyze and use these media. Timothy Lenoir and Sha Xin Wei describe both an extensive multimedia publication on the history of Silicon Valley and the underlying system that supports it. In his article, Alex Pentland, head of perceptual computing at the M.I.T. Media Laboratory, discusses current and future technologies for organizing, searching, and presenting images. Robert Gray describes more specific retrieval techniques based

on color and edge.

Additionally, limits of technology make it difficult to store multimedia data and to deliver it at appropriate rates. James Storer describes data compression techniques that facilitate the transfer of multimedia data, and Chetan Gopal and Roger Price describe delivery mechanisms specifically tailored for multimedia information. A promising use of multimedia is the presentation of information in multiple modalities. T. V. Raman explores this idea with an electronic publishing system that reads documents aloud to the visually impaired and provides corresponding non-visual navigation mechanisms.

Electronic publishing is making a significant impact on our notions of copyright, and both the legal system and the legal community are working to understand the implications electronic publishing has for intellectual property protection laws. In addition, there is an active interest in extending these laws to accommodate the changes wrought by electronic publishing, and in continuing to ensure that both authors and readers maintain sufficiently broad rights. Glen Secor and Carey Heckman provide an introduction to these issues, and Steven Ketchpel and John Erickson suggest mechanisms that protect authors and readers.

Electronic publishing is likely to have a broad social impact in the next decade. While it may promise greater access to all, it can also hinder access for those not literate in the technology, those without easy access to networks or sufficiently powerful machinery, or those unable to pay the promised "reduced costs" for electronic publications. Barbara Simons, chair of the Association for Computing Machinery Public Policy Committee, discusses these possibilities in the context of the forthcoming National Information Infrastructure. While these technologies may seem convenient to some, they are both confusing and overwhelming to others. Adrienne GreenHeart describes issues of control and content of multimedia, and who it is and should be designed for.

Electronic publishing is also having a significant impact on education. Many students are now able to better learn concepts through interactive demonstrations and through electronic access to expert teachers (or the materials from expert teachers). Edward Murphy, president of PWS Publishing, presents a publisher's perspective on the changing face of academic publishing. Viera Proulx, Richard Rasala, Harriet Fell, Peter Gloor, Marian Williams, and Johannes A. G. Koomen discuss the uses of animations in education. Networks may be instrumental in changing the way we learn; Dr. Joseph Henderson describes new, network-based learning models and how they are like and unlike traditional ones. Many authors, including Albert Henning and Leslie Bondaryk, present experiences creating educational materials.

Because academic researchers often have access to more or newer technologies, they have been able to take particular advantage of electronic publishing mechanisms to better disseminate their results. Peter J. Denning, chair of publications for the Association for Computing Machinery, describes the ACM's plans for publishing its journals, conference proceedings, and other documents electronically. Samuel Rebelsky, Robert Allen, Frank Baker, Charles Owen, and Robert Mack describe experiences creating electronic proceedings for a variety of conferences, which require quick turnaround but have the advantage of a large author community. Of additional interest is the so-called "electronic journal," which provides the trappings of a traditional journal—refereeing, standards, consistent formatting—in a completely electronic publication. Hermann Maurer and Klaus Schmaranz describe the *Journal of Universal Computer Science*, which includes its own hypertext system, and J. Redi and Y. Bar-Yam describe *InterJournal*, an electronic journal that incorporates material from electronic documents that can be widely distributed physically. Two panels, one chaired by Donald Kreider and one by John James, describe further issues in electronic journals from the perspectives of publishers, authors, and librarians. Academic publishing involves more than journals, and may require cooperation across many departments. A group from Southern

Illinois University describes the ground-breaking Freedom of the Press Project, an electronically published archive of censorship activity, developed by researchers, librarians, and the university press.

While electronic publishing has recently achieved mass acceptance, there have been electronic publications available for many years. Andries van Dam, creator of one of the first hypertext systems, describes the past, present, and future of electronic publishing from the perspective of one who has been creating electronic books for over twenty-five years. Electronic publishing is also not the only revolution publishing has undergone—past revolutions include movable type, the printing press, and even alphabetic writing. Gregory Crane will suggest lessons these past revolutions suggest for electronic publishing.

Finally, electronic publishing promises large "digital libraries" of electronic documents. If these libraries are to be useful, they must structure information appropriately and provide mechanisms for retrieving and updating information. Daniela Rus, James Allen, Klaus Süllow, Rainer Pagé, John Buford, Sara Elo, Kristen Summers, and Robert B. Allen describe a variety of systems, structures, and issues that are important in supporting digital libraries and the efficient and easy retrieval of electronic publications.

The summaries above cover only a fraction of the many ideas and developments described in these proceedings. We expect that you will find yourself challenged by new ideas, provoked by new directions, and stimulated to begin, continue, or extend your own work in electronic publishing.

James Ford
Fillia Makedon
Samuel A. Rebelsky

Note: Since it may seem odd that the proceedings for a conference on electronic publishing is in printed form, we feel we should stress that there are still many advantages to the printed medium. For example, readers can easily annotate pages, writing on top of and next to text and pictures, using text, mathematics, figures, colors, and whatever else is appropriate. We are constructing an electronic version of the proceedings that will be published by Addison Wesley Interactive after the conference, and we invite you to follow our progress at `http://www.cs.dartmouth.edu/ dags/DAGS95/Proceedings`.

The Dartmouth Institute for Advanced Graduate Studies (DAGS) and The Dartmouth Experimental Visualization Laboratory (DEVLAB)

The Dartmouth Institute for Advanced Graduate Studies was founded in 1991 by Donald B. Johnson and Fillia Makedon to provide an environment in which theoretical computer scientists, experimental computer scientists, and users of computing technology could come together to discuss problems and solutions. The DAGS symposia give theoreticians the opportunity to learn about more practical problems that they might attack from a theoretical perspective, experimentalists an opportunity to learn more about underlying theories and potential applications of their ideas, and users an opportunity to further explore the technologies that support their work. Initially, the DAGS institute focused on parallel computation as a way of bringing these communities together.

The first symposium of the institute was held in 1992 with a topic of "Parallel Computation: Practical Implementation of Algorithms and Machines." Speakers at the institute included Guy Blelloch, Charles Leiserson, Andrew Ogielski, Tom Leighton, Marco Annaratone, Gary Sabot, and Charles Van Loan. An interactive electronic proceedings, including presentations and papers, was published by Springer-Verlag. The second DAGS symposium, held in the summer of 1993, focused on "Parallel I/O and Databases." Speakers at the institute included John Wilkes, Jeffrey Vitter, David Waltz, Marina Chen, Alok Aggarwal, and David Scott. A new, cross-platform, electronic proceedings was developed for this institute but is not currently available. The third DAGS symposium, held in the summer of 1994, was on problem solving environments and "Providing Massively Parallel Computing to Problem Solvers." Speakers included John Reif, Constantine Polychronopoulos, Elias Houstis, and Elaine Kant.

The institute has also worked on training new researchers and practitioners. Each institute provided fellowships and scholarships for undergraduates, graduate students, and recent Ph.D.s. Each institute also included an accompanying school on issues from parallel computing, parallel I/O, and problem-solving environments, respectively.

The Dartmouth Experimental Visualization Laboratory (DEVLAB) provides further training for young scientists. Founded in 1991 by Fillia Makedon, the mission of the DEVLAB is to apply computer technology to a variety of uses, including education, and to give undergraduates and graduate students experience working with multimedia and visualization techniques. The DEVLAB supported the construction of the electronic proceedings for the DAGS institutes, using undergraduate interns to develop interfaces and manipulate materials. Other DEVLAB projects include VideoScheme, a programmable video editing system; WISKIT, a multimedia "kit" to attract women to computer science; the electronic classroom; a collection of multimedia visualizations for teaching parallel computing to novices; and multimedia information retrieval research.

This year the institute is expanding its focus and collaborating with the DEVLAB, and has selected electronic publishing as a topic. This is an active and exciting field, one with many interesting aspects for researchers to study. It is also one of the most broadly applicable uses of computing technology today. Just as prior symposia brought together several communities, the current institute brings together computer scientists, publishers, authors, librarians, commentators, readers, and the many other groups involved in electronic publishing. Our invited speakers, authors of research papers, and panelists provide a range of ideas and perspectives on electronic publishing.

INVITED TALKS

The ACM Electronic Publishing Plan And Interim Copyright Policies (Summary)

Peter J. Denning
Chair, ACM Publications Board

For the past three years, the ACM Publications Board has been developing its vision for the future of publication in the electronic age and a program to achieve it. We envisage a diminishing role for print journals and exciting new programs around an ACM digital library and new copyright practices. As we move aggressively into electronic publishing, we will preserve and extend the traditional openness of ACM publications in the new media. Authors and readers should find the new framework at least as hospitable as the traditional one.

Publishing has reached an historic divide. Ubiquitous networks, storage servers, printers, and document and graphics software are transforming the world from one in which only a few publishing houses print and disseminate works, to one in which any individual can print or offer for dissemination any work at low cost and in short order. This poses major challenges for publishers of scientific works and for the standard practices of scientific peer review.

The scientific publishing tradition, in which ACM founded its publication program, has two central tenets. The first is that manuscripts are published only after careful and deliberative review by experts. Not only is it considered wasteful to publish a paper that contains errors or repeats earlier work, it is an affront to the tradition of science to publish statements easily refuted by experts. The second tenet is that every published paper is a permanent member of the library of all scientific literature. In this tradition, a journal paper passes through the four phases of preparation, review and revision, publication processing, and archiving and indexing. In the new practices that are arising in the Internet, the moment of publication occurs with posting on a home page and is much sooner than the moment of imprimatur given by an editor who accepts the paper after successful review. Authors invite comments on their papers posted in this manner, and will often produce improved versions after official publication.

Although less visible, the policies and practices of archiving and indexing are as critical as publishing. A society's imprint would be worthless without reasonable assurances that the published work will be preserved for posterity and that readers can locate the work without having to locate the author. Even though digital libraries will offer new possibilities for archiving and indexing, the responsibility of the society to assure the archiving and indexing of its own authors' materials will not diminish.

ACM has a broad range of programs to disseminate information in various forms to people who can use it. ACM's publication program includes the traditional journals of scholarly research ("track 1"). It includes magazines and other services specifically designed to communicate with those who develop computing hardware, software, and services ("track 2"). ACM's strategy consists of six parts:

1. Maintain the health and vitality of the traditional "track 1" research-oriented publications by constantly repositioning them, by introducing new ones, and by offering access electronically. (Ongoing.)

2. Acquire and process all manuscripts electronically into SGML format for storage in the library database and for rapid translation into printer's codes. (Operational by summer 1995.)

3. Establish tools that support review processes so that the turnaround time will be under 2 months and the reviewing load of any given individual does not exceed that person's capacity. (Experimental versions operational by end 1995.)

4. Expand the line of "track 2" publications for practitioners and developers. (Ongoing.)

5. Establish the ACM Digital Library, an on-line database of ACM works. (First version available by mid to late 1996.)

6. Experiment intensively with prototypes of new services.

The ACM digital library positions ACM to offer new services that will make ACM members differentially more competitive than nonmembers. Over time, ACM expects to realize most of its revenue from three principal businesses:

1. Guided access to literature through search and access to the ACM Digital Library. Individual members can be notified of new entries matching their profiles. Nonmembers can obtain access licenses or pay per item retrieved.

2. Conferences, including new forms on Internet.

3. Continuing education through professional knowledge certificate programs. These programs are designed to make available the information donated by authors for further education of wider groups of people.

Copyright Policies

Under the traditional ACM copyright policies, documents were considered as property whose value had to be protected by release fees and permissions. Authors did not mind transferring copyright to ACM, since ACM was an author's principal agent for bringing material to readers; copyright transfer was a reasonable price to pay for dissemination. ACM allowed authors to retain the right to reuse any portion of the work in a future work with only a proper citation to the ACM published version.

Many of these assumptions are changing in the Internet. It is cheap and easy to copy an electronic document and thus nearly impossible to protect an electronic file as property. Authors are beginning to post manuscripts on their home pages and servers, making it seem to them that publishers are less relevant as agents of dissemination; the primary functions of a publisher now are to give an imprimatur of quality to a work, to find more readers than the author might find alone, and to maintain archives of ACM works. New situations not anticipated in the original policies have posed new questions. One is whether posting a document on an Internet server constitutes prior publication and thereby disqualifies the author from submitting it to ACM for publication. Another is whether an author who implants a hyperlink to a copyright document is effectively incorporating that document and needs to get the copyright holder's permission.

The ACM Publications Board has developed and issued a set of new copyright policies. They are labeled "interim policies" because they are subject to review and revision as we gain experience with them. As in the past, the ACM will hold the copyright on items it accepts for publication; this allows it to freely disseminate on the author's behalf without having to check with the author in each instance, and it will protect the ACM digital library from expropriation by those who might attempt to reproduce the same service for free. The principal assumptions are:

1. ACM grants authors liberal rights, including the right to reuse the copyright material in any future work provided that proper citation of the copyright work is given, and the right to post preprints and revisions on home pages for noncommercial purposes and personal use by others in the Internet.

2. ACM assumes that most revenues will come primarily from value added services such as database access, conferences, and professional knowledge certificates. Copyright release fees will not figure significantly in business plans.

3. A link to another document is treated as a citation. It is a matter between the individual attempting access via the link and the copyright holder what fee, if any, must

be applied. An author does not have to obtain permission to include a link.

4. Anyone obtaining an ACM copyright object may use that object only for personal use unless explicit permission has been granted for other use. This limits third parties (excluding authors) from distributing ACM materials freely in the Internet without ACM permission.

5. Servers from which ACM copyrighted objects can be downloaded must display to browsers a general notice advising that copyright materials are posted here and are subject to copyright limitations specified by their individual holders.

References

Denning, Peter J., and Bernard Rous. 1995. "ACM Electronic Publishing Plan and Intermim Copyright Policies." *Communications of ACM* (April), p97ff. These documents plus and author's guide are available on the server `<http://www.acm.org/pubs>`.

Biography

Peter J. Denning is Chair of the ACM Publications Board. He was formerly the editor-in-chief of the *ACM Communications* and was president of ACM 1980-82. He is associate dean for computing in the School of Information Technology and Engineering at George Mason University, where he is also chair of the Computer Science Department and Director of the Center for the New Engineer.

Copyright Notice

Electronic Books: Past, Present, and Future

Andries van Dam
L. Herbert Ballou University Professor
and Professor of Computer Science
Brown University

Abstract

Books have many attractive features: compactness, an easy-to-understand linear structure, relative permanence, and the opportunity they offer for annotation (marginalia, etc.). On the other hand, they are static, and therefore cannot exhibit dynamic properties: their pictures are fixed, and their content and presentation cannot be changed to suit the reader's needs and interests. The linear structure of books is also a constraint and does not correspond well to nonlinear organizations of knowledge. Finally, while an individual book is portable and cheap, a library of books is unwieldy, difficult to access, expensive to maintain, and even more expensive (and slow) to update.

Electronic books have the potential of preserving the attractive properties of books while ameliorating their shortcomings. A model for such books, and indeed of a digital library, that is becoming increasingly popular is that of a hypermedia database: a cross-linked collection of multimedia documents containing text, dynamic graphics, video and audio. The World Wide Web on the Internet is probably the best known example today. An essential ingredient of electronic books is user control of the presentation's content and format. An important example of such interactivity is what I call the "interactive illusion": user-controlled, real-time animation derived from stored models of physical and abstract objects and phenomena. Electronic books with interactive illustrations will be used for such applications as teaching and learning, entertainment, research, technical documentation and even electronic shopping.

Libraries, Old and New, and the Possibilities of the New Technology

Gregory Crane

Tufts University

This conference provides a forum in which we can speculate, one hopes constructively, on the impact of emerging digital libraries on the humanities. Attempts to predict the future by rational means are, however, hardly new. The fifth-century philosopher Demokritos is quoted as saying that (fr. 119) "human beings invented the image of chance (tuchê) as a pretext for their own foolishness, for only rarely does chance conflict with intelligence. Intelligent careful observation make most things in life run smoothly." Thucydides took a darker view. Perikles makes his debut in the history by citing the unexpected vicissitudes of human events (1.140). The history itself begins by advertising its utility for future generations (1.22.4). By the time Thucydides introduces his detailed account of the plague at Athens, his expectations seem to have declined, for he specifically states that he has no remedy or even helpful course of action to offer. The best he can offer is a case study so that future generations can recognize a resurgence of the same disease (2.48) — and, presumably, prepare to die. Even in this Thucydides was not successful: identifying the disease which devastated Athens in the fifth century has developed into a perennial scholarly pastime. In book three, Thucydidean confidence makes something of a comeback. In his analysis of civil war at Corcyra (3.82-83), Thucydides lists the atrocities and moral degradation of factional fighting. Here, he predicts that the events which happened at Corcyra and, indeed, all over Greece were not isolated but would repeat themselves so long as human nature remains the same. Certainly, the history of the twentieth century — with its parade of death squads and massacres, great and small — lends strength to the Thucydidean claim.

Optimism sells, but pessimism seems to wear better, at least in academic circles (thus Thucydides' history survives intact while we must content ourselves with fragments of Demokritos). The same tensions reflected in Demokritos and Thucydides are alive today as we all of us attempt to make sense of technological change and prepare for its future developments. Two recent books take up the Demokritean and Thucydidean sides of the argument. Nicholas Negroponte's *Being Digital* [Negreponte 1995] celebrates the dawn of a new age — indeed, of the post-information age — where bits replace atoms. Sven Birkerts, on the other hand, gives voice in his *Gutenberg Elegies* [Birkerts 1994] to the unease of the nervous intellectual. He frets about the death of literature and the fragmentation of the human self in a chaos of digital stimuli.

These are neither the best nor the worst recent books relevant to his subject,[1] but they form a neat pair and are in some ways the constitute a useful starting point of discussion.[2] On the one hand, the two authors come from complementary backgrounds. Nicholas Negroponte founded the fabulously well-funded Media Lab, and he currently seems to have assumed the role, formerly held by Marvin Minsky, as the foremost combination of scientist, visionary and snake-oil salesman at MIT (different observers will naturally attribute differing shares to these three roles). His book celebrates not only the on-going triumph of the new digital world, but Negroponte's own first-class journey through life, from his days in an expensive Swiss boarding school to his present

[1] Perhaps the most interesting recent book is Lanham 1993, which situates the current debate about new electronic media firmly in its intellectual context and allows the reader to gauge the impact exerted by the two thousand years of intermittent controversy. Stoll 1995 is an entertaining defense by a prominent hacker of traditional humanist values. Many of the objections that Stoll raises about digital media (e.g., that they isolate people in a world of their own) can and have been raised about books ("book-worm", "mere book learning" etc.). Stoll's book is interesting as a cultural phenomenon because it takes for granted such decidedly recent technological phenomena as the fax machine and because it often treats CD ROM and books together, as if they were equally traditional tools.

[2] I am hardly the first person to remark on this: see, for example, Jonathan Franzen's review of both in the *New Yorker* 71.2 (March 6, 1995) 119ff.

familiarity with the rich and powerful. Sven Birkerts, on the other hand, is a well-known essayist who earns his far less glamorous living across town from Negroponte, teaching Harvard freshmen how to write. His book talks not only about the corrosive effect of fast-paced interactivity on that deep quiet which nurtures the self but also about the author's own marginalized position as a writer of books in a television and now multimedia age.

Each book lends itself to caricature — indeed each is a very personal response that invites caricature. Negroponte points out the ironies inherent in publishing a book about the end of books as we know them. Birkerts, on the other hand, has produced not so much a book as a collection of essays which return again and again to the same theme: the form in which he writes — rapid prose, no footnotes, quick thumbnail descriptions, ideas succinctly packaged for easy and swift consumption — seems out of sync with the tranquil concentration and deep reflection on which he lays such stress. Both authors are disarmingly frank about their attitudes towards the new technology: Negroponte has done hugely well as a prophet of change, and he wants more of it; Birkerts bemoans the declining status not just of literature but of authors such as himself. Where Negroponte revels in his access to the great, Birkerts specifically laments the fact that writers like himself have ceased to command such attention.

I would, however, like to stress one element common to both Negroponte and Birkerts and to which classicists are in a particularly strong position to respond. Both authors seem to share a sense of technological determinism. Both assume not only that electronic technology is changing — and inevitably changing — our world (something which few would dispute) but also that the direction of change is relatively clear. The attitudes of these and other analysts, in some ways, recall the initial enthusiasm of Marshall McCluhan [McLuhan 1962], Jack Goody [Goody and Watt 1963], and the classicist Eric Havelock [Havelock 1963] when, in 1962 and 1963, each after his own fashion located in writing an inherent logic that inevitably reshaped the human mind. Subsequent study — in which classicists have played an important role — has not reinforced the strong hypotheses which these men advanced, for the interaction between society and writing is more complex than these earlier studies supposed: William Harris [Harris 1989], Barry Powell [Powell 1991], Rosalind Thomas [Thomas 1989; Thomas 1992], Deborah Steiner [Steiner 1994], and even Havelock's admirer Kevin Robb [Robb 1994] have all in recent years articulated the ambiguities of writing in the archaic and classical Greek world. Writing, like movable print, may be an agent of change, but it could, like print, also be turned to the service of the *status quo*.[3]

The more complex view of writing's influence deserves close attention for those who wish to assess the impact which electronic technology will have, for, if media do have inherent logics, human beings have their own agendas and do not always follow the path of least resistance. Detailed case studies about the influence of writing and of print should play a prominent role in the debate about contemporary developments. Negroponte makes no pretensions to a deep love of history or literature — indeed, he openly proclaims himself a dyslexic who does not enjoy reading — but it is discouraging to find in Birkerts, who claims to be defending a proud tradition of literature, such a shallow sense of what literature is. His model of authors and of texts concentrates primarily upon novels and short stories as they took shape in the nineteenth century. While he laments the passing of the traditional literary canon, his book evinces little familiarity with that intellectual breadth and variety which even the much maligned Dead White European Male authors of the traditional Canon embody.[4]

But these more nuanced recent analyses do have one weakness that earlier work did not. So much emphasis is placed on continuity and the hypotheses are often so weak that we find ourselves little the wiser about how and why things did change — for writing really was an agent of change, as was print two thousand years later and as is electronic technology today. It is simply easier from a methodological perspective to debunk strong hypotheses about the

[3]On the printing press as an agent of change, see the monumental and classic study: .Eisenstein 1979; for a collection of more conservative essays, see Hindman 1991.

[4]He includes, I might add, an essay on the Perseus database which he had first published in *Harvard Magazine*. The essay had, however, little to do with this database and its actual strengths and weaknesses but took used the database as a starting point for more general ruminations about technology and society.

inherent logic and impact of media than it is to determine, within the welter of contradictory evidence, the fuzzy, but real, path of historic change. Different media really do have different strengths and weaknesses, and these strengths and weaknesses in turn tilt the field towards some functions and away from others. The momentum behind human practice is, however, enormous — people do things the way that they are accustomed to doing things and they don't change their habits lightly, especially if they have been successful with the traditional scheme of things.

Consider Birkerts' claim that "cyberspace" will destroy literature. It would be easy, as many of those who have commented on his book have, to dismiss this, and certainly electronic technology will not destroy artistic expression, but literature in its most literal sense depends upon letters and writing. For some of us who have worked on the Homeric epics and see in them the climax of an ancient tradition of oral poetry, it is easy to imagine a world filled with poetry but without "literature." If we want to understand what might happen to literature, we could do worse than to consider how the continuous tradition of European literature emerged in the first place.

It is all too easy to push useful questions about Greek oral poetry too hard: were the Homeric poems composed with the aid of writing? Did Greek oral poets memorize things or did they always compose in performance? How much of the Homeric epics are formulaic? What precisely is a formula? Each of these questions is important, but none has been answered in any way that has compelled broad acceptance — a good sign that the questions should not be overworked.[5] Nevertheless, there are a number of things that we can say about Greek oral poetry that do bring out its peculiar nature.

First and above all, the Homeric epics are composed in a formulaic system that was designed to facilitate the production of hexametrical poetry. The poets who mastered this system were, in effect, fluent in a dialect of Greek which added metrical rules to the normal grammar and syntax of the language: these poets could thus speak in hexameters the way some people can speak in a foreign language or in an abstruse dialect. There is, in principle, nothing to prevent them from memorizing stretches of poetry — clearly, the rhapsodes who performed Homeric epic did just this, as did every educated Greek of the classical period to some extent.[6] The point is not that memorization is impossible but that memorization defeats the purpose of that extensive training which makes such memorization unnecessary: for an oral poet to memorize stretches of his work would be a little like turning your (compact) car into a horse-drawn wagon or using your expensive new shoes to hammer nails. Memorization defeats the purpose of, and runs contrary to, a magnificent formulaic system which had evolved over more than a thousand years.

Second, the formulaic system is, in its own way, extremely economical. By this I mean that if the oral poet had a serviceable formula to express a common idea — the rise of the sun at dawn or one hero addressing another — then he will use that formula as often as it is convenient without feeling the need to alter it for the sake of variety. Of course, oral poetry can be monotonous and good poets will not hammer the same formulae into the ground. But the formulaic system clearly did not prevent the poetry of the *Iliad* and the *Odyssey* from expressing a staggering range of ideas and developing characters of great subtlety. Those who mastered the formulaic system could, like the masters of any developed language, make it express anything that they chose. They simply did not feel the need to change language to conform with modern conventions of variety.

Third, the formulaic system extends beyond the phrases which constitute individual lines and into larger story patterns. Oral poets have basic frame-works with which they can describe a standard feast, a sacrifice, a one-on-one duel or similar type-scenes. Odysseus' encounter with the man-eating giant Laistrygonians is a compact version of a general story which finds fuller expression in the adventure with Polyphemos. Bernard Fenik analyzed what might be called the syntax of the Homeric battle scenes, and I began my scholarly career by examining the recurring

[5]There is a huge and specialized literature on Homer and oral poetry; Nagy 1990 17-51 and Thomas 1992 29-51 provide a introductions to this topic; two classic accounts (each with a very different perspective) are Lord 1960 (more recently, Lord 1991) and Finnegan 1977.

[6]Plato's dialogue, the *Ion*, is about a rhapsode and provides one very readable glimpse of these performers.

patterns which underlay such mysterious women as Calypso, Circe, Helen, and Nausikaa.[7]

Fourth, the regularity of this oral system contributes to its peculiarly expressive power. Precisely because the poet never needs to change his formulae or story patterns for the mere sake of variety, subtle changes can convey powerful messages. In the language of information theory, a message contains information insofar as its contents differ from what one would expect. The regularity of formulaic composition establishes a clean background of expectations against which variation stands out. Thus, on the level of the scene, the death scenes of Patroklos and of Hektor follow much the same pattern, and the similarity invites comparison. Patroklos' final words, for example, warn Hektor that his death is near, but Hektor refuses to accept this, and the poem represents him arguing with a Patroklos who is already dead (Il. 16.843-861). When Hektor dies, he too warns Achilles that his fate is approaching. Achilles, however, grand and terrible in his rage, has long since accepted his fate and makes no attempt to wriggle free (Il. 22.335-366). The regular frame highlights the difference between the futility of Hektor and the terrible majesty of Achilles. Or, consider the effect of regular patterns in epic tradition as a whole: immortality with the goddess Calypso on the beautiful island Ogygia has much to commend it, but Odysseus' possible fate is not simply attractive. It belongs to a conventional pattern in which the hero escapes death, finds a paradise on the edge of the world (either Elysium or an "island of the blessed") and has as his consort a beautiful divinity. This pattern, precisely because it has its own independent existence, throws into greater emphasis Odysseus' decision to leave. Odysseus has reached the point for which heroes are expected to strive. When he gives up immortality and his blissful existence in favor of a mortal wife and eventual death, he violates set expectations and dramatizes his attachment to his home.

The Homeric poems exploit the strengths of oral composition. Even if we withhold aesthetic judgments, dismiss their merits as poetic achievements, and fall back upon the language of our social scientist colleagues, the Iliad and the Odyssey can be defended on objective grounds as two of the most successful artistic productions of any culture. They have fascinated millions of people from many different cultural backgrounds for more than two thousand years. They continue to sustain both a general audience and scholars who devote decades of concentrated work. They constitute an extraordinary monument to human creativity. Their objective success suggests that each poem was the product of a single unifying intellect. The inconsistencies of the poems, such as they are, were not flaws but reflect a different set of aesthetic conventions (or, to put it in terms appropriate to a conference on digital libraries, "those aren't bugs — they're features"). Perhaps these poems evolved in the hands of many different poets, but I find it hard to believe that they would have enjoyed such astonishing success for so long. The first poems preserved in European literature have been matched but arguably never excelled.

The Homeric poems represent a transitional stage from the oral to the written. They are the products of an oral tradition, but they have been preserved as written texts. They are, in fact, a bizarre hybrid, for there is no good evidence to believe that traditional Greek oral poets composed such monumental epics. Greek oral poets clearly focused, like Phemios and Demodokos in the *Odyssey*, on episodes that could be recited at a single sitting. Greek practice conforms with Poe's dictum that a single poem must be heard in a single sitting — anything longer than this breaks down into many smaller poems. I have no idea whether the monumental poet(s) of the *Iliad* and the *Odyssey* were literate, or whether these poems were dictated. Arguments that the complexity of these compositions requires writing are problematic, because, like Perikles' Athenians (2.35.2), they measure the possible by the capabilities of themselves and their acquaintances. Whoever composed these poems was, in some part of their minds, as different from the rest of us as Shakespeare and Mozart — who, it might be noted, composed almost entirely in their heads and wrote down their work when it was virtually complete. On the other hand, the *Iliad* and the *Odyssey*, although products of an oral poetic tradition, needed the fixity of writing if they were to survive. Their enormous length ran counter to the instincts of oral poetic tradition. Some have argued that Greek writing was even invented to record these

[7] Crane 1988; Fenik 1968.

works,[8] and the evidence supports this hypothesis as well as it does any other.

But if writing saved the Homeric epics for posterity and indeed made such massive poems feasible, writing also killed the oral poetic tradition out of which these poems emerged. The death was doubtless slow, but, by the classical period, the thousand year tradition of oral Greek hexametrical poetry, which predated the emergence of the Greek language and had begun evolving as Indo-European, had died out forever. Writing clearly undermined the need to master the complex formulaic system that we can see in the Homeric epics as well as in Hesiod and the Homeric Hymns. The focus shifted from rapid composition to the perfected written script that was memorized and performed. Athenian drama was, in a way, oral literature, for it was designed for performance, but it was a written literature, in which actors and chorus practiced extensively to master fixed parts. New media do often have different internal logics. These logics may take time to exert themselves. They may not be deterministic and can thus evolve, like species, in different directions. But a new medium with new strengths and weaknesses presages fundamental change. Anyone who thinks that any intellectual structures of our culture — whether the book or film or television — is fixed should contemplate the fate of Greek oral tradition which lasted more than a thousand years but evaporated after the appearance of writing.

I think the probability very high that electronic media constitutes a new medium as novel as were writing and print. Indeed, the long term consequences may be even greater, more comparable to the development of human language itself, for, bit by bit, we have begun to store sophisticated skills in abstract form on which we can call. No machine can analyze natural language and perhaps none ever will, but machines can recognize all the forms of the Greek verb *ferô* or generate three dimensional models of ancient structures. Whether our colleagues in computer science ever develop artificial intelligence in a general sense, we can already begin to see more limited, but cumulatively powerful, "tools of Hephaestus" — electronic aids that perform many small, often complex, tasks.

No one knows where all of this will take us. If the past is any guide, change catalyzed by technology has no set, deterministic course of development. Nevertheless, if we humanists are unlikely to get what we want, we have a better chance of getting what we deserve and what we deserve depends upon the depth in which we ponder the relationship between what is now possible and our real goals.

To this end, I would like to stress one principle, hardly surprising in itself but the implications of which are often overlooked. Those of us who are humanists must strive to make our assumptions explicit and in this degree must pursue scientific objectivity. Likewise, those of us who design reference works — whether print or electronic — have much in common with architects and other builders of useful things. Nevertheless, we in the humanities are not scientists and we are not engineers. I stress this point because, to the extent that we locate the center of our field in our research and the production of knowledge, we accept a logic of evaluation in which we will always be second class: we are not pursuing a cure for AIDS; we are not building a human GENOME project that may unlock the secrets of the human body; we are not physicists seeking to build the theoretical foundation for superconductors or controlled fusion; we are not even sociologists struggling to trace and describe rising crime or drug use, or cognitive scientists who methodically expand our understanding of the human mind. If we place our research first, then we embrace a game that we can never win and in which we are not needed. And I say this as a classicist whose favorite pastime is in fact research — disciplined, systematic, long-term study.

Nevertheless, our research plays a indispensable role in our work and contributes to, even if it does not constitute, the value of our field. The humanities naturally pursue a modified labor theory of value — our work only has value to the extent that it commands the attention and interest of those outside of our professional ranks. We have many tasks — to challenge prevailing assumptions, to keep alive the many pasts which make up our collective identity, to hold up a mirror for our fellows in which they can see their own lives in the distinct hues of different peoples and cultures, and, yes, even to challenge and expand tastes beyond the buzz of film and television and even the cliquish brilliance of the New York Times Book Review. The materials that we study —

[8] Powell 1991.

whether they are ancient Greek texts or modern post-colonialist fiction — have value only insofar as they earn serious attention, now or in the future. To this end, our research plays a crucial role, whether by articulating theories which allow us to see with new eyes or by providing background that brings the culturally or historically obscure into sharper focus. Nor can this process ever end, for, if we are stimulate questioning and an active intellectual life, we must ourselves lead the way.

The greatest danger that confronts us, as humanists evaluating the new technology, is to think only in terms of the old audience. New tools that help practicing researchers do their work more effectively will always win some support, but we, as humanists, can never be content — and certainly not now — with reproducing in enhanced digital form the same scholarly conversation. Electronic tools can easily become even more specialized, abstruse and difficult to use than their print counterparts, and they can thus narrow even further their audience. In my thirteen years of continuous work in humanities computing, I have consistently found that electronic tools, properly designed, that serve the general audience can support research more effectively than their narrower counterparts.

Let me give one concrete example from the work of my own immediate colleagues in the Perseus Project. Ten years ago, I gave talks at Chicago and UCLA about the prospects of placing the major scholarly Greek-English Lexicon into electronic form. In the meantime, we worked with a student version of this lexicon, making it a part of the Perseus database and using it to familiarize ourselves with the problems of electronic lexica and text bases. We spent several years of work building an intelligent, rule based morphological analyzer for classical Greek because we knew that we would be able to solve two logistical problems: non-specialists would be able to go from inflected forms to dictionary entries — not a trivial task when a single Greek verb can have, when preverbs are considered, literally millions of forms, and the form on the page need not resemble the dictionary entry; specialists would be able to go in the opposite direction, starting with a dictionary entry and locating in a text all of its inflected forms — again a non-trivial task. We found, however, that a simple index of the English definitions of the dictionary turned the Greek-English Lexicon into a very powerful English to Greek dictionary. Furthermore, by exploiting the tight links between Greek words, dictionary and text, even those with no knowledge Greek were able to locate all passages in which a key term, e.g. *aretê*, occurred and then scrutinize the way in which this was translated. In other words, the technology allowed us to make accessible problems buried in the Greek to an entirely new audience.[9]

In the spring of 1995, we have finally succeeded in completing the preliminary data entry of the large Greek lexicon. While we have much work to do, we can already see one major phenomenon that we had not properly anticipated. In this case, we can easily make the electronic text much more readable than the dense, small type of the print lexicon: we can increase the point size, put blank spaces between definitions, use bold print to highlight dictionary entries, and fill out obscure abbreviations. If we are reading a particular author, we can scan the definitions in a long article those which cite what we are reading. We can even trivially compile an index of all cited passages: there are, for example, more than 8,000 citations of 1,000 chapters in Thucydides. In famous sections — such as Perikles' Funeral Oration or the description of the Plague at Athens — dozens of words on each page are specifically cited in the lexicon. Where the big lexicon had, because of its size, been too cumbersome for any but the advanced student, the electronic lexicon seems even more "friendly" to the general student than the abridged dictionaries which they have used for over a hundred years. A moment's reflection should make clear the huge advantages — intellectual as well as economic — to having a single central reference tool used by everyone from intermediate students (of whom there are many) to faculty (of whom there are few).

I would like at this point to shift my focus from the past to the present and future. I do not wish to oversimplify or to imply that it is possible for us now to predict, with any precision, the consequences which these new digital libraries will have or even the forms which they will assume. Nevertheless, I do think that we have learned some things, and I would like then to address several common misconceptions that seem to cause widespread, if often unexpressed,

[9] This topic is covered, with illustrations of screens, in Crane 1991.

anxiety among my colleagues. What I have to say will perhaps be of most immediate interest to those in the humanities and social sciences, but conversations with my colleagues in the natural sciences suggest that their problems are not as different as we all may think.

First, cynicism is easy. "There is a sucker born every minute" and "no one ever went broke underestimating the taste of the American public" are famous sayings. It is easy for us to become discouraged when we spend our days teaching often distracted, frequently overworked undergraduates, or constantly struggle with parents and administrators anxious that students immediately receive highly paid, secure jobs: the professor teaching organic chemistry to hordes of pre-meds suffers just as much as the professor of classical Greek anxiously maintaining his enrollments. But cynicism does not account for all the phenomena. There is an enormous reservoir of intellectual curiosity and energy waiting to be tapped. Whenever I get discouraged, I think of my wife's grandfather, who was both a farmer and a lay preacher in the Methodist church, and who over the course of a long, hard life mustered the energy to work his land and to scrutinize his bible in an exacting detail that would put many of us philologists to shame. Then I try to imagine whether a twelfth century priest, ministering to this man's illiterate and malnourished ancestors, would have believed that such people could one day master the complexities of the bible. When we imagine broadening our audience, we do not need to restrict ourselves to a least common denominator. Even television — which remains for the most part a cultural wasteland that seeks numbers above all — has begun to reflect the broad based curiosity of our fellow citizens: the new cable networks abound with a growing number of programming on scientific and historical topics. The content remains crude, and the tendency to oversimplify, inherited from traditional broadcasting, has a momentum that will take some time to dissipate, but these materials point the way to new developments. Things do change. Writing cost us Greek oral poetry but it gave us Sophocles and Plato. The grossly centralized and bureaucratic mass media of the twentieth century are no more permanent.

Richard Lanham, whose book, The Electronic Word [Lanham 1993], exemplifies perhaps better than any other recent publication the realist optimism that I am trying to express, describes multimedia as the revenge of the book on television. Let me stress that technology can develop in very different ways depending upon the values and conscious decisions of a society, but consider, for example, the following scenario.

At the present time, there exists a tremendous gap between mass media and scholarly media. The Mass media are, we all know, deeply problematic. They are capital intensive — it takes tens of thousands of dollars to produce even simple "broadcast quality" material. An hour of standard video programming for the networks has costs that start in six figures and rise, for the most elaborate shows, to seven. The result is a horrific centralization of creative control and the evolution of imperial cliques ranging from the stars of the newspaper checkout magazines to the narrator for the PBS Series Nova, who has his own company to market his widely recognized voice. Let there be no ambiguity: this is a terrible state of affairs, the smiling consumerist version of a totalitarian state. The few — the very, very few, for even your traditional elites tend to constitute at least one percent of the population — construct what we see and hear. The many change channels, shifting perhaps to the Learning Channel or to such canned products as Time or Scientific American.

By contrast, let me praise the traditional scholarly publications. The players here require extensive training, the teaching positions which facilitate research are scarce, and research materials are, at least for now, tied to a university libraries. Nevertheless, if you have the training and access to a good library, you don't need to get a grant to write an article or maintain a studio with fixed costs of $1,000,000 a week to study women writers in the eighteenth century. It is even possible for someone without a Ph.D. to contribute to academic research. Our colleagues in astronomy, for example, know this, for they have long depended upon a network of disciplined amateurs for crucial information — the man who discovered Pluto sixty years ago was, for example, self-taught. To the extent that the complexities of modern astrophysics have pushed such people aside, the field has been diminished. Even in classics, with its arcane and often technical infrastructure, non-specialist participation is possible. Robert Strassler, for example, a businessman from the Boston area, became fascinated in Thucydides. He has published at

least two articles on the Pylos campaign in the Journal of Hellenic Studies — a leading publication in the field — and he is finishing an extraordinary edition of Thucydides for a major publisher, filled with extensive maps and a series of authoritative essays by experts in the field. Of course, this degree of participation is unusual, but publication is only dimension of the serious intellectual life. The subscriber to the History Book Club can do very well intellectually, for this institution represents a rare synthesis of mass media and scholarly publication, for it offers widespread access to a variety of outstanding, thoughtful texts.

The strength of the History Book Club highlights a weakness in our traditional scholarly infrastructure as a whole. Mass media are ubiquitous — indeed, some American outlets flood the globe: the sun never sets on CNN and MTV, the Lucy Show lives on, dubbed into a farrago of languages that would shame the tower of Babylon, even the Wall Street Journal is transmitted electronically and then printed for every major financial center. Scholarly publications are, by contrast, the peasant farmers of the information world, with limited physical range and less influence over affairs. Let me be blunt on this: the center for academic information, the print library, occupies a role very much like that of the medieval clergy. Remember that in a world where most are illiterate, manuscripts hugely expensive, and labor cheap, a professional priesthood is probably the most efficient and progressive mechanism whereby to disseminate knowledge. When printed books arrive and a far wider audience can gain access to writing, then the professional priesthood, which had been proud of its contributions to society, faced new challenges for which it was not prepared. Electronic media — which can travel around the world at the speed of light — have begun to push the book as artifact — the book as a physical object that can be only in one place at a time and can only serve readers sequentially — into a position as ambiguous as the medieval priesthood. But this analogy is hopeful as well as challenging. If the medieval church brought the reformation on itself, the catholic school systems of the nineteenth and twentieth centuries command admiration for their contributions to democracy. If agents of the Catholic Henry the Eighth kidnapped William Tyndale, dragged him to England and burned him at the stake for publishing the first English translation of the bible, the Catholic Archdiocese of Boston supports the only independent schools in Boston financially accessible to a wide segment of the population.

We can now imagine a new synthesis, one that can unite the strengths of the mass media and of scholarly publication. Will books disappear? Not soon. The more pertinent question might be: will manuscripts — by which I mean handwritten documents — and handwriting itself disappear? The point that I wish to stress is that a printed book is now the physical representation of an electronic entity that is as superior to the printed artifact as Plato's forms are over physical reality. This is not mere rhetoric: the printed has a tangible existence that we may cherish, but its form is defined electronically. The printed book may sit securely on the library shelf for a hundred years, but its stability is a prison, a death row, in fact, for any serious scholarly work, for the printed text cannot be updated. If we wish to add new information, remove outdated material or revise sections we must turn to the ethereal electronic version. With this we can generate printed documents in many formats or publish material on the Internet.

Nor have bytes subsumed print alone. At long last, the digital tools at our disposal have begun to eliminate the fuzzy, evanescent and permanently expensive analog media. Video is a miserable medium: coarse to begin with, it loses quality rapidly. It has little to defend it, although any institution has its defenders. Even film, however, has begun to give ground — whatever Jurassic Park's merits as an artistic creation, the digital dinosaurs constituted an "existence proof" that showed what digital media could do. Let me emphasize what this means. A good scanner can extract at least twenty megabytes from a single 35 mm slide before the grain of the film begins to show up. If the engineers can keep making machines cheaper, faster and more capacious for just a few more years, film itself may be in trouble.

Whether we read a new book of poems in quiet solitude, gaze at the flickering small screen in our homes, or watch films in their more communal setting, bytes have become the *lingua franca* — perhaps *media franca* would be a better phrase — for all things published or broadcast. This fact, although often hidden behind the deceptively stable forms of books, videos and movies, makes possible the new, heterogeneous multimedia. Creative authors have

already begun to exploit emerging possibilities. Consider only one example. Robert Winter's hypertextual edition of Beethoven's Ninth Symphony, published in the late 80s by the Voyager company, may sound exotic, but its approach was commonsense: Winter was able to link his words to subsections of the symphony. He made his audience readers *and* listeners at once. He made accessible to a new audience the systematic, interactive, guided scrutiny of music which had previously only been available in the classroom. His historic publication has spawned a new genre and proved decisively that the new media was not incompatible with the best goals of traditional culture.

We now live in a world in which we can begin to break down the stultifying barriers between the mass and the scholarly media. Imagine a video in which you can not only select which segment you wish to view, but can then freeze the video and find your way into the sources and explore a library. All of us who are speaking here today at this conference are helping to build an infrastructure that can lead seamlessly from the most glossy and elegant staged video presentation to the most abstruse and rigorous scholarly tools. As I said at the beginning, no one can predict the future and technology limits but does not determine where things can go. Change may take decades, but our students and our children will live to see a world vastly different from that in which we grew up. The extent to which this is a better world remains largely in our hands.

References

Birkerts, S. (1994). The Gutenberg Elegies. Winchester, MA, Faber and Faber.

Crane, G. (1988). Calypso: Backgrounds and Conventions of the Odyssey. Frankfurt am Main, Athenäum.

Crane, G. (1991). "Composing Culture: the Authority of an Electronic Text." Current Anthropology 32: 293-311.

Eisenstein, E. L. (1979). The Printing Press as an Agent of Change. Cambridge, Cambridge University Press.

Fenik, B. (1968). Typical Battle Scenes in the Iliad. Wiesbaden, F. Steiner.

Finnegan, R. (1977). Oral Poetry, its Nature, Significance and Social Context. New York, Cambridge University Press.

Goody, J. and I. Watt (1963). "Alphabetic Culture and Greek Thought." Comparative Studies in Society and History 5(3): 42-54.

Harris, W. V. (1989). Ancient Literacy. Cambridge, Harvard University Press.

Havelock, E. (1963). Preface to Plato. Cambridge, Harvard University Press.

Hindman, S. L., Ed. (1991). Printing the Written Word: The Social History of Books, circa 1450-1520. Ithaca, Cornell University Press.

Lanham, R. A. (1993). The Electronic Word: Democracy, Technology, and the Arts. Chicago, University of Chicago Press.

Lord, A. (1960). The Singer of Tales. Cambridge, Harvard University Press.

Lord, A. (1991). Epic Singers and Oral Tradition. Ithaca, NY, Cornell University Press.

McLuhan, M. (1962). The Gutenberg Galaxy: the Making of Typographic Man. Toronto, University of Toronto Press.

Nagy, G. (1990). Pindar's Homer: the Lyric Possession of an Epic Past. Baltimore, Johns Hopkins University Press.

Negroponte, N. (1995). Being Digital. New York, Alfred A. Knopf.

Powell, B. (1991). Homer and the Origin of the Greek Alphabet. Cambridge, Cambridge University Press.

Robb, K. (1994). Literacy and Paideia in ancient Greece. New York, Oxford University Press.

Steiner, D. (1994). The Tyrant's Writ: Myths and Images of Writing in Ancient Greece. Princeton, NJ, Princeton University Press.

Stoll, C. (1995). Silicon Snake Oil: Second Thoughts on the Information Highway. New York, Doubleday.

Thomas, R. (1989). Oral Tradition and Written Record in Classical Athens. Cambridge, Cambridge University Press.

Thomas, R. (1992). Literacy and Orality in Ancient Greece. New York, Cambridge University Press.

MMDD and Networked Scholarly Workspaces

Sha Xin Wei
Academic Systems Development
Stanford University
Stanford, CA 94305-3090
415-725-3152
xinwei@jessica.stanford.edu

Timothy Lenoir
Department of History
Stanford University
Stanford, CA 94305-3090
415-725-1524
tlenoir@leland.stanford.edu

http://www-leland.stanford.edu/~xinwei/
http://www-leland.stanford.edu/group/STS/lenoir.html

This talk presents MMDD —a framework for composing distributed media, in the context of the SiliconBase project. SiliconBase is a research project in the history of Silicon Valley, conducted by members of the Program in the History and Philosophy of Science. The MMDD mediates between network multimedia services and interface kits with which application programmers may easily fashion radically different interactive views into shared mediabases. The network services include search engine abstractions, filters, relational modeling frameworks. Faculty and student authors compose distributed media using Macintosh, NeXTSTEP and World Wide Web applications, supported by services from common UNIX workstations.

Introduction

A major challenge facing designers of networked computing environments today is to fashion scholarly workspaces which are simultaneously
coherent,
easily reconfigurable,
expressive —small gestures go a long way,
and above all, *worth using*.

In this talk, we'll describe the Metamedia Distributed Databases(MMDD) —a system for composing arbitrary renderable media, applications, or mediastreams in diverse models and narrative structures. The MMDD is designed to support the construction of models of human systems which are both conceptually rich and data rich. It also mediates between coherent, customizable interfaces and an open set of network services, such as database engines, WWW servers, fulltext search engines, and media conversion facilities. (See the Gallery of MMDD applications[1].)

Our context is humanities computing [Thaller], which significantly stretches the envelope of networking technology, multimedia, intelligent search systems, and human-computer interface design. Software technology paradigms now run the gamut from verb-object tools (set the color of the selected word to red) to document processing, intersubjective computing and urban design. [Alexander] We take a perspective situated somewhere between urban design and intersubjective computing. Our method has been to have designers/programmers work intimately with the faculty and student researcher/authors who use the evolving systems. [Ehn] In fact, the MMDD was conceived in the beginning as a framework to accelerate our own multimedia designers' work in creating rich complexes of media supported by relational data models. It was natural to extend the notion of designer to include authors who were experts in fields outside computer engineering.

History

After about five years of making interactive multimedia titles, we took stock of our work process to see where the bottlenecks were, and also what were the greatest defects in the interactive titles which we produced.

- Media were scattered all over the network. It was becoming hard to keep inventory using ad hoc databases.
- Researchers significantly changed their conceptual models over the course of a project, so that custom data structures had to be re-written.
- User interfaces had to be constantly re-designed in concert with graphics artists, programmers and researchers, using unpredictably varied media. New interface constructs such as help sprites and custom gestures which did not fit pre-fabricated window-menu-button widgets had to constantly invented.
- Finished titles were often locked into a videodisc or piece of software (eg. Supercard stack), and put out of reach of re-purposors.
- Finished titles had thin media content/ hard content boundaries —users quickly hit the boundaries of what was recorded on a CD ROM or videodisc.

[1] http://lummi.stanford.edu/Media2/pix/www/MMDDScreens.html

- Conceptual models were often too simplistic to be taken seriously by any but the most novice students. We wanted environments which could support research level work as well as introductory classes. (In general, software which was designed specifically for a given class or lesson was often too rigid and shallow.)
- Hypertext/media graph topologies were either navigable but too sparse to sustain a viewer's interest, or rich but too dense to be comprehended. Traditionally, hypertext links are fragile, difficult to author or manage, and hard to map.
- We could not easily support multi-author and multi-player discourse networks.

The MMDD was designed to address all of these problems. Its various frameworks were designed to be used by faculty and student authors and by designers of multimedia simulations; it was designed explicitly to support members of academic disciplines outside traditional programming communities. And it had to leverage tiny application programming resources.

We started with two prototype projects in 1993-1994: a history of Renaissance (Elizabethan) theater, and a study of high technology in the Silicon Valley. The first was chosen from a pool of faculty projects which required some management of art images and associate music or text on the network, The second presented the challenge of dealing with a significant, changing body of structured text in a complex, evolving research model. In addition, we wanted to lay the foundation for general relational modeling of human systems as such data became available in the course of the research. In both cases, we could not assume a fixed interface or conceptual model. Indeed, the only surety was change. This genealogy strongly influenced the design principles which we will outline in the following section.

Since then we've continued with the SiliconBase [Lenoir], as the Silicon Valley History project is called, and have added several other communities + mediabases: a prototype for an archive of electro-acoustic music; a Chicana/o artists database; and most recently, the Information Map Project which aims to serve as both a learning center and a clearinghouse of Latin American conservation issues and organizations.

Design Principles and Corollaries

Make it immediately useful.

Bread & butter reasons, but also participatory design principles suggested that we should let composers start working right away with their own media, conduct seminars and write papers using our system instead of waiting for the Holy Grail. To enable significant scholarly work, whatever we built had to exchange data transparently with commercial applications and databases, and inter-operate transparently with distributed services. Authors were encouraged to use whatever commercial editors they already had on their personal computers (Macintosh, some IBM PC): eg. MS Word, WordPerfect, Adobe Photoshop, Adobe Premiere, Omnipage, DeskScan. Our frameworks synthesize commercial, public and custom software. Our authors work in a heterogeneous network where UNIX and Mac clients see a common filesystem, and can apply user tools from Mac, UNIX/X and UNIX/NS to shared mediabases.

Factor, factor, factor.

The architecture reflects a separation between (1) **persistent storage** in the filesystem (eg. ASCII or AIFF blob bytes) and in databases (eg. blob metadata in Sybase tables); (2) **model** (eg. hypermedia topological structure, bibliography); and (3) **presentation/ interaction** (eg. WWW/Mosaic document, Hypercard simulation, custom disposable apps). By decoupling models from media, we can sidestep the question of data ownership and allow complex research models to be constructed on existing corpora or proxy media. [1]

Since the MMDD stores topological information in databases, it can generate HTML documents on the fly rather than keep source media in HTML files —a simple version of dynamic documents. Factorization gives us the option of interposing even more expressive and nuanced means of forming constellations media or mediastreams *on the fly*.

Maintain user interface metaphor neutrality.

We wish to allow multiple views on shared media, which means that rather than building a single interface application or layout protocol (*a la* HTML forms), we provide an API supporting multiple, concurrent, and most importantly, reconfigurable interfaces. The MMDD does *not* assume that views must look like word-processors. Word-processor-like document viewers like MS Word or Mosaic present essentially a unidimensional rebus, a stream of generalized characters, some of which are ordinary letters, some of which are raisins of media like an embedded graphic. In general, a simulation can have quite a different structure, such as a map, timeline, multi-track score, vivarium, video VR, soundspace etc. MMDD user interface kits do not assume documents, windows, chunks, or links. But the MMDD does deliver documents as a special case. For example, ordinary

word-processor documents may be catalogued in indigenous formats.

Broadcast rather than publish.

The MMDD is designed to deliver information over networks, rather than in detached forms such as CD ROM. The CD ROM (and videodiscs etc.) distribution model is in a sense a natural relic of the traditional publishing model which requires a physical commodity in order to function. From the point of view of a university library, most if not all of the same problems encountered in acquiring preserving, cataloguing and circulating paper books or journals recur in dealing with CD ROMs and videodiscs. Some of these library issues are even thornier in the new formats.

Finegrained network distribution of software, even of single computing objects, offers quite a different paradigm which may be more akin to a broadcast model than to the publishing model. This also gives us the flexibilty we need to support live research projects in which the primary source media as well as the secondary literature and even the conceptual models are in flux. In any case, the MMDDs factorization allows us to build templates to which we can download a subset of a projects model + data at any moment. In this way, we can print a standalone version of simulations like T. Gieryn's Cornell Biotechnology Lab or G. Crane's Perseus by downloading data and models from the network into local templates.

Even more interesting are the new genres of publication now made possible by online mediabases. The MMDD provides a scheme in which progressively more formal or public compositions can arise organically from flexible, personal or project-specific research collections. For example, collections of source material can be acquired and edited according to research agenda. This demand-driven model efficiently allocates human and system attention. New scholarly articles or pedagogical presentations can be made *in situ* and catalogued back into the mediabase. For example, the SiliconBase seminar's reader is an entirely online hypermedia structure which can be modified at any moment by the instructors. Lectures can be composed, presented in conferences, and revised online. Over time, well-critiqued articles can then be given more public status by relaxing their access locks. Such research reports become a virtual professional journal with the addition of a suitable editorial board and digital signatures. Design issues such as the social conventions around periodicity and cost recovery mechanisms would be interesting to investigate using such a framework.

Maintain model neutrality.

To allow multiple conceptualizations requires that authors be able to build rapidly several models over the same media. This came from a practical need to reconcile the very different time-scales involved in designing provisional research schema of annotations and associations vs. designing a MARC-quality archival description of the same set of media. Again, by factorization and abstraction the MMDD allows very different communities to work with media, represented when necessary by proxies, using their own models. Consequently, instead of binding to one particular database, the MMDD uses a data access framework which allows us to connect to any of several standard types of RDBM engines over the net, including Sybase and Oracle. The MMDD provides an object-oriented abstraction so that its clients need not deal with dialects of RDBMs. Clients can store arbitrary objects like bitmaps or serialized Objective-C objects as meta-data via the MMDDs object-oriented database access framework. In practice, (large) media are kept as source media in ordinary distributed filesystems, and (small) meta-data — annotations, references, links, abstracts, etc. — are kept in RDBMs.

Expect evolution.

Perhaps the key to making an scholarly workspace worth using is to ensure that intellectual content survives across change in technology. This is partly an institutional commitment as well as a technological issue. Aside from the obvious requirement of a modularized architecture whose components may be replaced without breaking service, the following principles guided our work:

Assume no single data representation.

We do not need to spend resources to converting media systematically to a single format like HTML or SGML. This is perhaps the most important technical feature of the MMDD. By making no assumption about the internal structure of a media entity (a blob), and not even requiring that a media entity exists as bytes in a filesystem, the MMDD allows authors to compose with any computable or renderable medium whatsoever. This way, the MMDD can accomodate currently unknown data types and interactions. Moreover, this way the MMDD can deal with opaque or pre-recorded media (eg. TIFF, MPEG, AIFF, TeX, Renderman), performable scripts (eg. NS scorefiles, Mathematica notebooks, Applescripts), executables (eg. a UNIX tool, Hypercard stack, NetScape application), and data streams (eg. live video channel) with equal ease /difficulty.

How is this feasible? The general principle here is to

Focus on space of transforms more than the base space.

Converting all the authors source media into some standard structure (such as SGML) is not cost effective nor strategic in our context because of the diversity of the material (some conversions would lose too much information), the large human cost (editorial, programmer, administrative), and the constantly changing substance. Moreover, we are not convinced that a universal permanent (on the scale of decades) document structure exists which can deal with all the structures we have in hand. Therefore, we decided that it is wiser to build a filter service which MMDD core objects as well as clients could invoke on foreign platforms.

Assume nothing about the internal structure of a media entity.

A media entity may be a programmatically generated stream of data, a file of any renderable data type, an executable, or may even exist only as a virtual object in a meta-data record. This allows authors to reason with proxy objects even when, for legal or technical reasons, primary media are not available. Conversely, multiple versions of a logical media entity can be tracked. The front end, not the MMDD, decides how to interpret multiple versions of a blob. For example, a movie clip may exist in MPEG as well as a QuickTime Mac proprietary format. The front end asks for the locally renderable version, but authors deal only with a single logical entity.

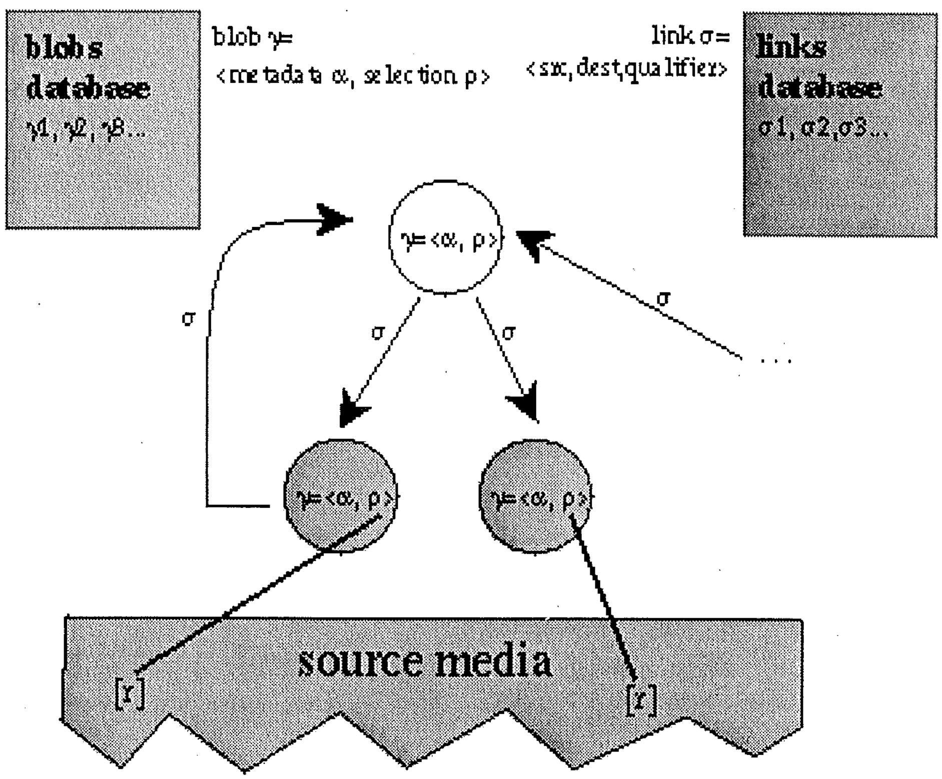


Figure 1: Media Model

Architecture

The basic media object model is described in Figure 1. Each media entity, or blob, has a unique tag, zero or more source versions, and zero or more attributes/ metadata fields/abstracts. Typically, the logical media entity is associated to some data ("source media") in persistent storage, but this is not required. By allowing virtual blobs which refer to no source media, we can construct compound structures quite naturally. The MMDD has four framework levels:

- a set of user interface kits (Macintosh, NS, WWW),
- a set of mediation class libraries (TCP-IP, WWW, general Service Object Manager, NS DBKit),
- a set of services (mentioned above), and
- persistent storage (AFS, AppleShare, Sybase).

Under the assumption that editors, browsers, search engines, filters, abstractors, and high-level OO inter-operable user environments could be added incrementally and in parallel, we invested more of our energy into the service mediation, plus abstract classes which captured the semantics of search, annotation, and association. In fact, the MMDD is now integrated with many of these complementary tools. For details, see [Sha2].

Where do we go from here?

Now that we have a sufficiently rich substrate of services, a small but diverse set of scholarly user communities and corpora, we would like to turn our efforts to make the user environments more seamless. In the near future, we would like connect MMDD front ends with commercial siblings such as GIS apps, and SM: or numerical engines. We are evaluating multi-architecture, metaphor neutral user interface frameworks which can talk to the MMDD. Kaleida's ScriptX is one possibility, as are Apple's OpenDoc and JAVA. [2]

We will be extending the project in several application areas, including relational models of human systems, and geographic information systems. Project disciplines include art, anthropology, history, literature and theater.

End Notes

[1] We have in mind notions such as using relational grammars to define meta-layouts for user-interfaces. Examples include WRI's Mathematica 2.3, and work by Weitzman and Wittenburg.

[2] Originally called Oak. WebRunner was written in Oak. James Gosling, jag@sun.com.

Bibliography

[Alexander]
Christopher Alexander, Sara Ishikawa, Murray Silverstein. *A Pattern Language.* Oxford University Press, 1977.

[Eha]
Pelle Ehn. "Towards a Philosophical Foundation for Skill-Based Participatory Design." *Usability: Turning Technologies into Tools*, 116-132 in P. Adler and T. Winograd (eds.) Oxford University Press, 1992.

[Lenoir]
Tim Lenoir, Sha Xin Wei.. "Networked Scholarly Workspaces for History of High Technology." Talk at MIT, March 1995, online document available at URL `http://lummi.stanford.edu/Media2/pix/www/MIT/slides_contents.html` 1994.

[MMDD]
"Metamedia Distributed Databases." Online document available at URL
`http://lummi.Stanford.EDU/Media2/ASD/ASD_Homepage/Multimedia.html` 1994.

[Sha2]
Sha Xin Wei, Deborah Zimmerman, Rick Wong, and Siew Sim. "MMDD - A Distributed Meta-media Simulations Framework." Submitted to *Multimedia Systems Journal.* November 1994.

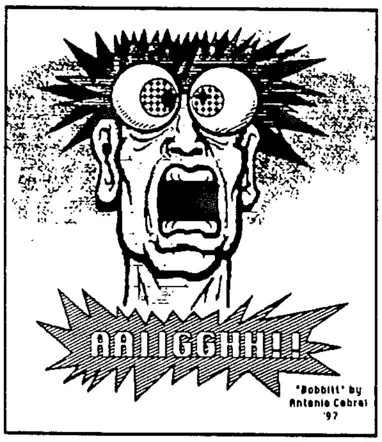
[Thaller]
Manfred Thaller. mthalle@gwdg.de. Max-Planck Institute for History, Goettingen, Germany. "What is 'source oriented data processing'; what is a 'historical computer science'?" In *Historical Informatics: an Essential Tool for Historians?* 1994.

[Weitzman]
Louis Weitzman and Kent Wittenburg. "Automatic Presentatiopn of Multimedia Documents Using Relational Grammars." *ACM Multimedia 1994.*

Multimedia Document Engineering for Nonmajors

Peter Wegner
Brown University

Brown's introductory course Concepts and Challenges in Computer Science (CS2), which has grown in size from 45 students two years ago to 85 last year and 180 this year, combines conceptual computer literacy with practical proficiency in the creative use of application packages. The MacPaint competition, assigned during the first week and judged by our team of undergraduate teaching assistants (UTAs), brings out interesting artistic talent. This year's 12 prizewinners (in the categories best artistic, best technical, funniest, and most original) and nine honorable mention entries are exhibited on the second floor of the Computer and Information Technology building. Two prizewinning pictures are shown below—last year's funniest picture (which turned out to be the star attraction at a number of public lectures on CS2) and this year's best technical prizewinner.



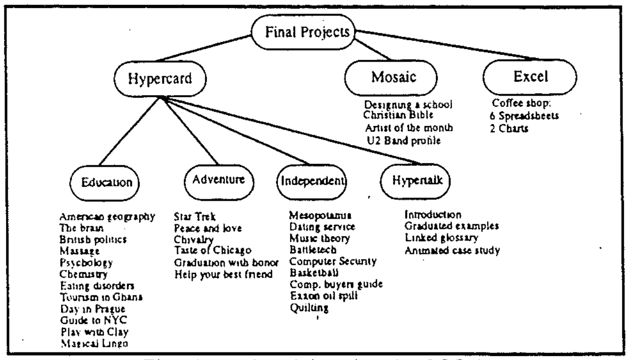
Funniest, 1994 "Bobbitt"

Best Technical, 1995

The course is assignment-driven with a simple Hypercard résumé assignment in the second week, a 12-hypercard assignment on "How Computers Execute Programs" in the third week, and a network assignment in the fourth week that includes a network treasure hunt, a simple HTML home page, and an essay on network architecture. By the fourth week students are familiar with MacPaint, MS Word, Hypercard, e-mail, and network surfing using a simple viewer like Netscape. They are encouraged to access the course home page, which contains information about the course syllabus and UTAs as well as a "Message of the Day" (MOTD) providing up-to-date information about assignments, exams, etc.

The artificial intelligence assignment "Can Machines Think?", given in the fifth week, is based on Turing's seminal paper, ongoing debate between scholars like Searle and Penrose, and Isaac Asimov's *Bicentennial Man*. Its "home card" (see Figure 1, page 2) has buttons for accessing background information, facts, "yes" arguments, and "no" arguments as well as content, help, and opinion cards. A "personal opinion" card from one of this year's assignments (Figure 2, page 2) shows the subtle reasoning this assignment can elicit. Later assignments include a spreadsheet in Excel, a coffeeshop cash-register program in Hypertalk that involves both interface design and simple programming, and an essay on the social impact of computers that this year allowed students to focus on the spring-1995 special issue of *Time* on cyberspace published just a week before the assignment.

The month-long final project requires students to develop a hypertext on a topic in which they are interested, such as a hobby, a course, or an artistic or athletic interest. They are encouraged to develop a multimedia document that includes text, static and dynamic illustrations, and some audio. A "design" developed in the first two weeks must be approved before the implementation can go forward to ensure that it is neither too ambitious nor trivial. Last year's projects included hypertexts on the archaeology of Mesopotamia with artifacts and historical discussion of each period, a tutorial on modeling with clay with modern examples of pottery, adventure games based on Star Trek and other themes, a New Testament hypertext on Matthew illustrated with Leonardo's *Last Supper*, a tourist guide to New York emphasizing night spots, and a music hypertext on the band U2 with audio clips (see figure below).



Final project topics in 1994

Students become "computer literate" in the first half of the course and deepen their ability to create multimedia documents in the second half. The midterm exam tests computer literacy as defined by a list of about 300 terms that students are expected to understand. Questions have the form: what are the similarities and differences between instructions and data? Binary and Roman numerals? Compilers and operating systems? The vocabulary list is available prior to the exam and evening help sessions devoted to answering questions about the "literacy vocabulary" are generally well attended.

One of the features of CS2, as with our other introductory CS courses, is its intensive use of UTAs. The 19 UTAs of CS2 provide 25 heavily used hours per week of consulting without which the intensive schedule of assignments would not be possible. During the final project period each UTA works directly with ten students on their final projects. The department's investment in 50 to 100 UTAs per semester not only benefits students but also develops a departmental *esprit de corps* and a sense of responsibility and belonging. The team of UTAs is generally quite diverse, including Indian, Pakistani, African-American, and Asian-American women and men. This year the CS2 UTA team includes the presidents of the Brown Islamic and Jewish societies. Over half of our majors serve as a UTA for at least one course.

CS2 has attracted attention both in industry and at other universities as a model for computer literacy. I was invited to talk on CS2 at Bellcore's "Electronic Document Delivery Conference (EDD-94)", at Apple Computer, and in Europe. My talk at a workshop on introductory courses in January 1995 at Harvard sparked interest at several colleges in using the approach and materials of CS2. We hope to develop an exportable version of the course materials this summer.

CS2 differs from first courses for majors in focusing on documents rather than programs. Hypercard, Excel, HTML, and MS Word may be viewed as tools for document management, while programs may be viewed as specialized documents. Students learn not only a set of general-purpose tools and concepts but also develop the ability to express themselves in a new medium. CS2 teaches technical writing and design skills for substantive multimedia documents that could not be written effectively without computers. Methods of document design in CS2 are similar to those of program design, but are applied to the domain of documents rather than programs. The final "capstone" project requires students to apply their writing, design, and document management skills to a new domain about which they are already knowledgeable.

CS2 has a conceptual and technical coherence, focusing on the technology of *document engineering* (the creation and management of documents), which has many parallels with the established technology of software engineering. Large programs and large multimedia documents have a similar structure and similar management problems. Both are linked structures of components: links associated with buttons are introduce much earlier in hypertext-based courses than the

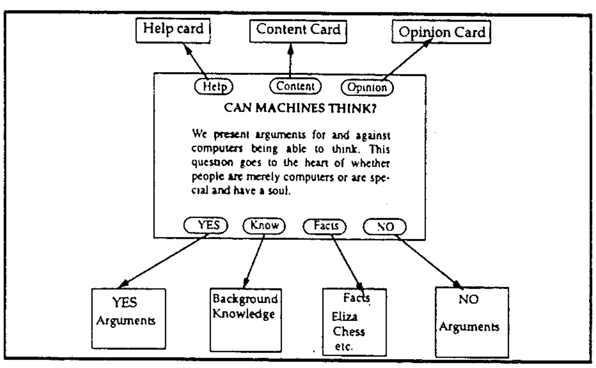


Figure 1—"home card"

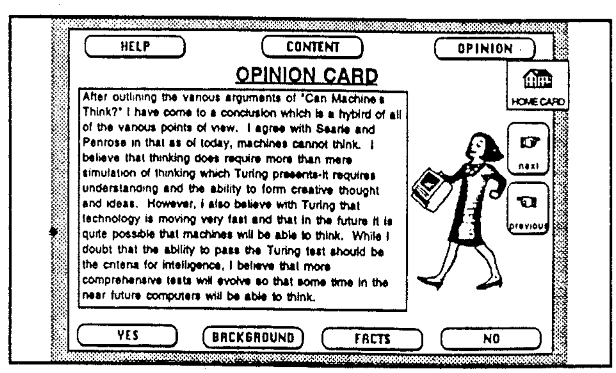


Figure 2—A "personal opinion" card

corresponding concept of pointers in Pascal-based or C-based programming courses. Large hypertext documents are easier to create in the time span of a single course than large programs; CS2 can explore problems of largeness without many of the technical details that arise in programming. Moreover, large documents relate move directly to everyday experience than large programs and more effectively motivate students to explore substantive large applications.

As computer science becomes more application-driven and outward-looking, first courses in computer science may evolve from their current emphasis on programming to an emphasis on document engineering. It may well be that ten years from now first courses on computing for majors will be closer in content to CS2 than to programming courses in Pascal, C, or C++. First courses in programming are becoming more design-oriented and less preoccupied with low-level algorithms and control structures. The gap between courses for majors and nonmajors is likely to narrow as the technology of personal computers matures and a conceptual framework that spans both programming and document engineering is developed.

High Performance Adaptive Compression

James A. Storer
Computer Science Department
Brandeis University
Waltham, MA 02254

Cornel Constantinescu
IBM Almaden
650 Harry Rd.
San Jose, CA 95120

Bruno Carpentieri
Dept. Informatica ed Applicazioni
Universita di Salerno
84081 Baronissi (ITALY)

Abstract / Introduction: We review some of our recent work on single-pass adaptive algorithms for the compression of images and video. The "theme" is to combine techniques from adaptive lossless text compression with quantization techniques to obtain algorithms that not only compress well but are single pass, highly adaptive, allow the compression - fidelity tradeoff to be continuously adjusted, and lend themselves well to high speed parallel / hardware implementation.

1. Adaptive Image Compression

Vector quantization is a powerful approach for lossy image compression when a good codebook is used, but the need to have this codebook supplied in advance can be a significant drawback. Constantinescu and Storer [1994a,1994b] show how to combine the ability of lossless adaptive dictionary methods to process data in a single pass with the ability of vector quantization accurately to approximate data. For a given overall fidelity of the decompressed image, the compression achieved by this new approach typically equals or exceeds the JPEG standard. In addition, it often out-performs traditional trained VQ (even in the best case, where the codebook is specifically trained for the type of data being compressed) while at the same time having a number of additional advantages: First, it is a single-pass adaptive algorithm (requiring no codebook to be provided in advance). Second, one can provide precise guarantees in advance on the distortion of any lxl sub-block of the image (whereas trained VQ simply finds the best match to an available codebook). Third, with a fixed codebook size, one can continuously vary the fidelity / compression tradeoff (whereas trained VQ typically achieves different tradeoffs by employing multiple codebooks). Our algorithm also enjoys some of the advantages of trained VQ, such as fast table-lookup decoding and fast parallel / hardware implementations for encoding.

1.1 The Basic Single-Pass Adaptive VQ Algorithm

With lossless adaptive dictionary methods, a local dictionary D is used to store a constantly changing set of strings. Data is compressed by replacing substrings of the input stream that also occur in D by the corresponding index into D; we refer to such indices as *pointers*. The encoding and decoding algorithms work in lockstep to maintain identical copies of D (which is constantly changing). The encoder uses a *match heuristic* to find a match between the incoming characters of the input stream and the dictionary, removes these characters from the input stream, transmits the index of the corresponding dictionary entry, and updates the dictionary with an *update heuristic* that depends on the current contents of the dictionary and the match that was just found. If there is not enough room left in the dictionary, a *deletion heuristic* is used to delete an existing entry. See the book of Storer [1988] for an overview of adaptive lossless dictionary compression and references to the literature.

Vector quantization is a lossy method that compresses an image by replacing sub-blocks by indices into a dictionary of sub-blocks. Traditionally, the sub-blocks are all the same size and shape and the dictionary must be computed in advance by "training" on sample data. Not only can training be computationally expensive, but "full search" encoding that is guaranteed to find the closest vector in the dictionary can also be very time consuming. In practice, tree-structured dictionaries are often used. Lin [1992] studies the performance - complexity tradeoffs for vector quantization. See the book of Gersho and Gray [1991] for an introduction to vector quantization and references to the literature.

Below are *Lossy Generic Encoding and Decoding Algorithms* for on-line adaptive vector quantization. Large rectangles are "grown" from smaller ones as the image is compressed. This process is depicted in Figure 1, which shows an image of a brain and a decompressed version of this image where each rectangle has been colored with a random solid color (to illustrate the covering pattern). It can be seen that in the easy to compress black areas around the image, very large matches were used, whereas smaller matches of varying size were used in the interior.

1. Initialize the local dictionary D to have one entry for each pixel of the input alphabet and the growing points pool (GPP) with one (or more) growing points.

2. Repeat until there are no more growing points in *GPP*:

   a. {Select the next growing point from *GPP*:}
   Use a growing heuristic to choose a growing point *GP* from *GPP*.

   b. {Get the best match block *b*:}
   Use a match heuristic to find a block *b* in *D* that matches with acceptable fidelity *image(GP,b)* (the portion of image determined by *GP* having the same size as b). Transmit $\lfloor \log_2 |D| \rfloor + 1$ bits for the index of *b*.

   c. Update *D* and *GPP*:}
   Add each of the blocks specified by a dictionary update heuristic to *D* (if *D* is full, first use a deletion heuristic to make space).

Generic Lossy Encoding Algorithm

1. Initialize *D* and *GPP* by performing Step 1 of the encoding algorithm.

2. Repeat until there are no more growing points in *GPP*:

   a. {Select the next growing point from *GPP*:}
   Perform Step 2a of the encoding algorithm to obtain *GP*.

   b. {Get the best match block *b*:}
   Receive $\lfloor \log_2 |D| \rfloor + 1$ bits for the index of *b*. Retrieve *b* from *D* and output *b* at the position determined by *GP*.

   c. {Update *D* and *GPP*:}
   Perform Step 2c of the encoding algorithm.

Generic Lossy Decoding Algorithm

The operation of the generic algorithms is guided by the following heuristics:

The growing heuristic: Selects one growing point *GP(x, y, q)* from the available pool *GPP*. All experiments reported here use the *wave* heuristic (a "wave front" that goes from the upper-left corner down to the lower right corner). Other examples of growing heuristics include *circular* (a "ball" that expands outward from the center), *diagonal* (a successive "thickening" of the main diagonal), and *FIFO* (first-in first-out).

The match heuristic: Decides what block *b* from the dictionary *D* best matches *imageGP* (the portion of the image of the same shape as *b* defined by the currently selected growing point *GP*). All experimental results reported here use the *greedy* heuristic (choose the largest match possible of acceptable quality, and among two matches of equal size, choose the one of best quality). The parameters that guide the matching process are: The *distance* measure; we use the standard mean-square measure in all experiments. The *elementary subblock size l*; large matches can be divided into subblocks of constant size *lxl*, and then distance is computed as the maximum distance among the subblocks; this prevents distortion from being unacceptable in a small portion of a match because it is better than needed in other areas (all experiments reported here use lxl = 4x4). The *type of coverage*; examples of image covering strategies include *first* coverage where the distance is computed only on the uncovered part of *imageGP*, *last* coverage where the match is computed for the entire block (except if it falls outside the image borders), and *average* coverage (used in all experiments reported here) where the match is computed for the entire block as for last, but on overlapped areas the resulting value is the average value between all the values of matches that happen to cover that pixel. The *threshold t*; a real number that defines the maximum allowed distance (distortion) between *imageGP* and *b*.

The growing points update heuristic: The growing points update heuristic is responsible for generating new growing points after each new match is made. For all experiments reported here, the concave corners of the partially encoded/decoded image are chosen.

The dictionary update heuristic: The dictionary update heuristic adapts the contents of the dictionary *D* to the part of the image that is currently encoded/decoded. All experiments reported here use the *OneRow+OneColumn* dictionary update heuristic that adds (if possible) two new blocks to the dictionary, constructed by extending the previously matched block (or part of it) vertically and horizontally by one row.

The deletion heuristic: Maintains the dictionary *D* so it can have a predefined (constant) size. All experiments reported here use the *LRU* heuristic (delete the entry that has been least recently used).

1.2 Experimental Results:

We used the following image test set (these images are shown in Figure 2):

ChestCAT: Cat-scan chest image, 512 by 512 pixels, 8 bits per pixel.

BrainMrSide: Magnetic resonance medical image that shows a side cross-section of a head, 256 by 256 pixels, 8 bits per pixel; this is the medical image used by Gray, Cosman, and Riskin [1991].

BrainMrTop: Magnetic resonance medical image that shows a top cross-section of a head, 256 by 256 pixels, 8 bits per pixel.

NASA5: Band 5 of a 7-band image of Donaldsonville, LA; the least compressible of the 7 bands by UNIX compress.

NASA6: Band 6 of a 7-band image of Donaldsonville, LA; the most compressible of the 7 bands by UNIX compress.

WomanHat: The standard woman in the hat photo, 512 by 512 pixels, 8 bits per pixel.

LivingRoom: Two people in the living room of an old house with light coming in the window, 512 by 512 pixels, 8 bits per pixel.

FingerPrint: An FBI finger print image, 768 by 768 pixels, 8 bits per pixel; includes some text at the top.

HandWriting: The first two paragraphs and part of the figure of page 165 of *Image and Text Compression* (Kluwer Academic Press, Norwell, MA) written by hand on a 10 inch high by 7.5 inch wide piece of gray stationary scanned at 128 pixels per inch, 8 bits per pixel; approximately 1.2 million bytes.

Table 1 shows three experiments for each image; each column shows to the left of the dashed line a given distortion expressed as SNR (or PSNR in parentheses) and to the right of the dashed line the compression obtained by our algorithm and by JPEG when both are set to achieve that SNR. Note that although our algorithm can be tuned to do better on a given image, the same set of parameters were used for all experiments (only the distortion threshold was varied).

	"Very Good"		**"Good"**		**"Fair"**	
	snr(psnr)	**ours/jpeg**	**snr(psnr)**	**ours/jpeg**	**snr(psnr)**	**ours/jpeg**
BrainCAT	29 (36)	4.3 /3.0	22 (29)	8.9 / 4.8	18 (25)	12.7 / 6.7
BrainMR_Side	28.5 (39)	4.1 / 4.6	26.5 (37)	4.9 / 6.1	20.5 (31)	10.3 / 15.8
BrainMR_Top	27 (35)	2.9 / 2.4	20.5 (28.5)	5.7 / 3.9	15.5 (23.5)	10.8 / 6.6
NASA5	30.5 (41)	4.2 / 4.1	28 (38.5)	5.6 / 5.9	26 (36.5)	7.5 / 8.5
NASA6	46 (51.5)	22.8 / 8.4	40.5 (46.5)	80.1 / 64.7	39 (45)	106.5 / 65.1
WomanHat	35 (40.5)	4.1 / 4.4	30 (35)	8.8 / 13.7	27 (32.5)	14.5 / 23.5
LivingRoom	32 (38)	4.0 / 4.3	27 (33)	7.5 / 9.1	24.5 (30.5)	11.0 / 14.3
FingerPrint	32 (35)	6.3 / 6.5	24 (27)	26.5 / 27.3	22 (25)	38.9 / 35.0
HandWriting	32 (33)	17.0 / 9.5	24.5 (25.5)	60.1 / 32.0	17.5 (18.5)	177.0 / 67.3

Table 1

1.3 Complexity Issues

Decoding involves little more than table lookup and is very fast. We have developed a KD-tree data structure for representing the dynamic dictionary of variable sized rectangles that speeds encoding time on a UNIX workstation from several hours per image to a few seconds per image with no significant change in compression of quality. A number of parallel / hardware implementations are also possible.

2. Adaptive Video Compression

A key component of many practical video compression systems (including MPEG based systems) is displacement estimation, where one is trying to "track" blocks of pixels from one frame to the next. Traditional displacement estimation algorithms used fixed size blocks. Here, we allow variable size and shaped regions of pixels to be tracked from one frame to the next. Pixels that move together are merged into a group called "super blocks" (which segment the frame); when pixels in a super block move in different directions they are split off from the block. Information about these splits along with corrections to regions of unacceptable error is primarily what comprises the compressed data stream. Information about motion in the image is implicit in the segmentation and split information and can be used to improve the error correction process.

Experiments have been performed with the following test set (Figure 3 shows sample frames):

Salesman: This standard test sequence (obtained by anonymous ftp at ipl.rpi.edu) consists of 448 frames, 360 by 288 pixels per frame, 8 bits per pixel. It contains relatively little detail or motion, typical of the head and shoulder sequences common in video telephone applications.

Fog: From the motion picture "Casablanca", the final scene when Humphrey Bogart and Ingrid Bergman say good-bye in the fog at the airport. This sequence is composed of 60 frames, 152 by 114 pixels per frame, 8 bits per pixel, digitized at a rate of 12 frames per second. There is a considerable amount of noisy movement due to the foggy background.

Kids: From the motion picture "It's a Wonderful Life", it's one of the first scenes, where kids (the main characters as children) are sitting at a desk. This sequence is composed of 100 frames, 152 by 114 pixels per frame, 8 bits per pixel, digitized at a rate of 12 frames per second. There is a fair amount of movement due to the presence of three characters.

Mountains: From the motion picture "The Sound of Music", one of the final scenes, where the main characters are walking in the mountains. This sequence is composed of 60 frames, 152 by 114 pixels per frame, 8 bits per pixel, digitized at a rate of 12 frames per second. The scene involve a noticeable amount of movement.

Pastorale: From the motion picture "Fantasia", a scene from the part of the movie illustrating Beethoven's 6th Symphony. This sequence is composed of 60 frames, 152 by 114 pixels per frame, 8 bits per pixel, digitized at a rate of 12 frames per second.

Table 2 shows the results we have obtained comparing our algorithm to a standard fixed size block, full search algorithm. The first column of the table identifies the sequence, the second column reports for each sequence the average SNR (in db) between consecutive frames as a measure of their correlation. The third and fourth columns present the results of the comparison between the standard algorithm and the Split-Merge algorithm for the test sequences. We have run the standard algorithm with block size 8 (8 pixels by 8 pixels blocks) and block size 4 (4 pixels by 4 pixels blocks) and we have reported in the first subcolumns of the third and fourth columns the average SNR between the original frames and the prediction obtained. Then we have run our algorithm setting the parameters in such a way to achieve that same average SNR and in the second subcolumns we have compared the size of the predictions; that is, the number of bytes needed to send the prediction from the encoder to the decoder assuming no lossless compression is performed. As can be seen in Table 2, for the same SNR, our algorithm has in general a noticeable saving in size respect to the standard fixed block, full search algorithm. In the sequence "Fog" the foggy background produces noisy effects on the segmentation performed by the Split-Merge algorithm, those effects are particularly relevant when we use a very small initial block size (2 pixels by 2 pixels). This is why the Split-Merge outperforms the standard algorithm in all experiments but in the case of the sequence "Fog" and initial blocksize 2.

Sequence	Correlation (Previous Frame)	Full Search bs8 vs Split Merge bs4		Full Search bs4 vs Split Merge bs2	
	SNR	SNR	SIZE	SNR	SIZE
Salesman	22.91 db	25.57 db	1822.5 vs 444.07	26.57 db	5670 vs 3733
Mountains	19.81 db	23.48 db	300 vs 138.96	24.92 db	931 vs 857,03
Fog	34,29 db	35.22 db	300 vs 141.06	36.94 db	931 vs 1248
Kids	25.87 db	27.59 db	300 vs 84.47	28.38 db	931 vs 695.58
Pastorale	22.79 db	24.12 db	300 vs 90.71	27.70 db	931 vs 893.20

3. References

C. Carpentieri and J. A. Storer [1994]. "Split-Merge Video Displacement Estimation", *Proceedings of the IEEE 82:6*, 940-947.

C. Constantinescu and J. A. Storer [1994a]. "On-Line Adaptive Vector Quantization with Variable Size Codebook Entries", *Information Processing and Management 30:6*, 745-758.

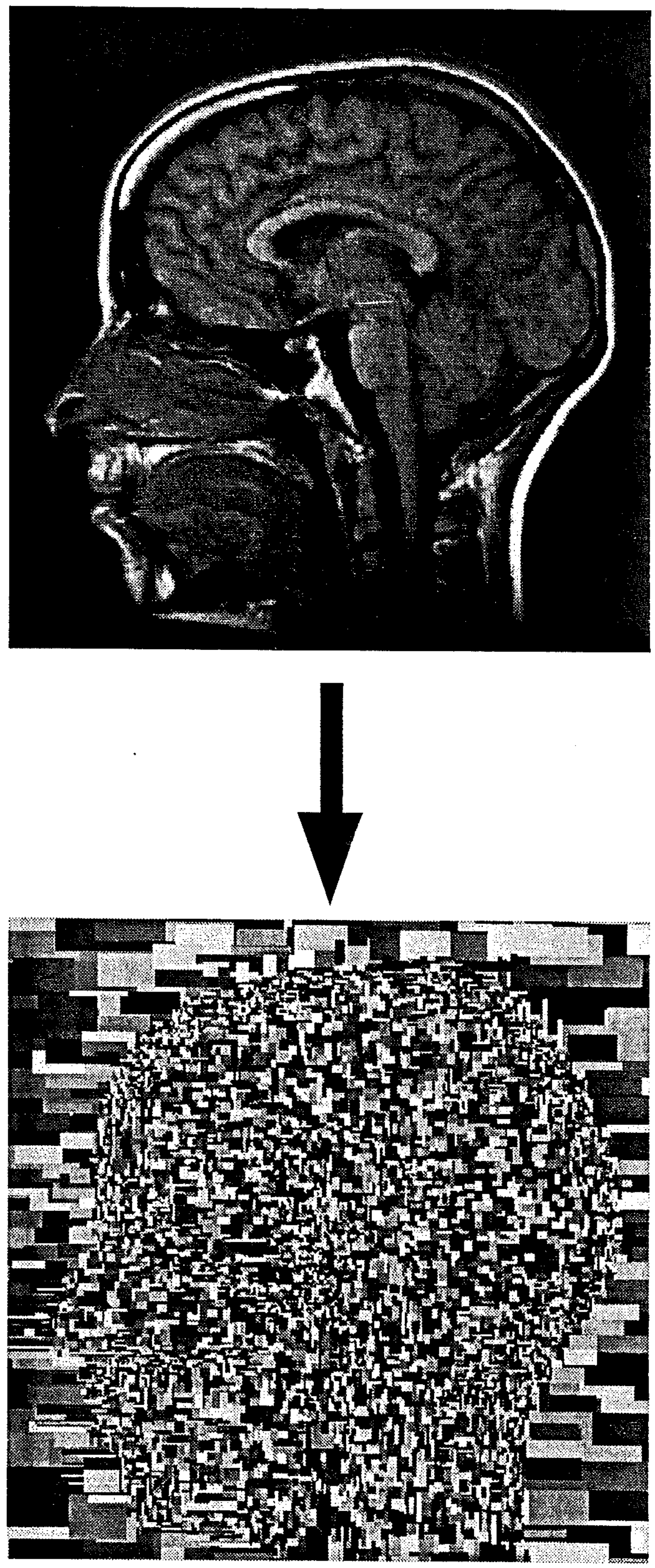
C. Constantinescu and J. A. Storer [1994b]. "Improved Techniques for Single-Pass Adaptive Vector Quantization", *Proceedings of the IEEE 82:6*, 933-939.

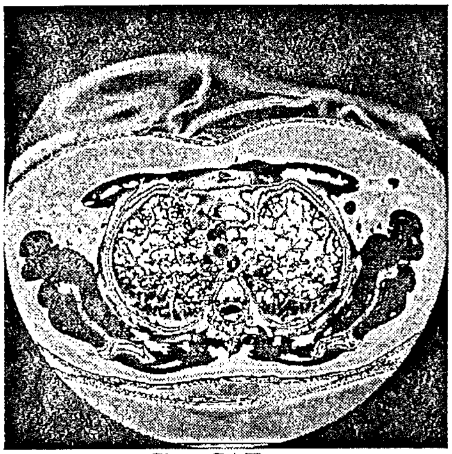
A. Gersho and R. M. Gray [1991]. Vector Quantization and Signal Compression, *Kluwer Academic Press*, Norwell, MA.

R. M. Gray, P. C. Cosman, and E. A. Riskin [1991]. "Combining Vector Quantization and Histogram Equalization", *Proceedings IEEE Data Compression Conference*, 113-118.

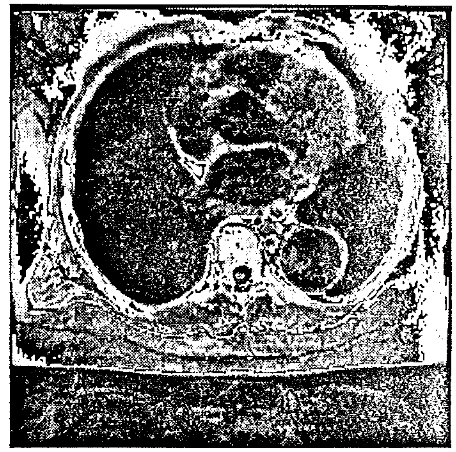
J. Lin [1992]. "Vector Quantization for Image Compression", Ph.D. Dissertation, Computer Science Dept., Brandeis University, Waltham, MA 02254.

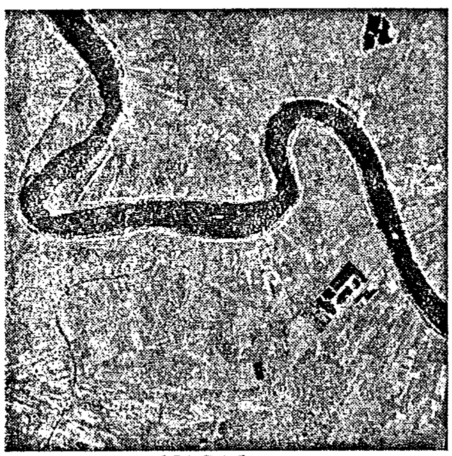
J. A. Storer [1988]. *Data Compression: Methods and Theory*, Computer Science Press, Rockville, MD.

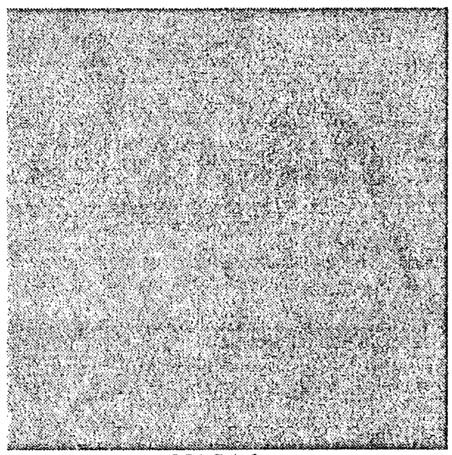


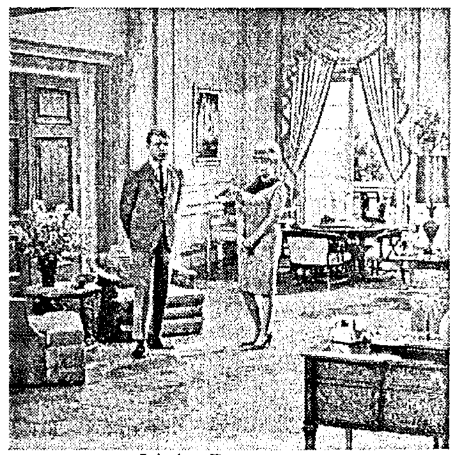

ChestCAT


BrainMR_Side

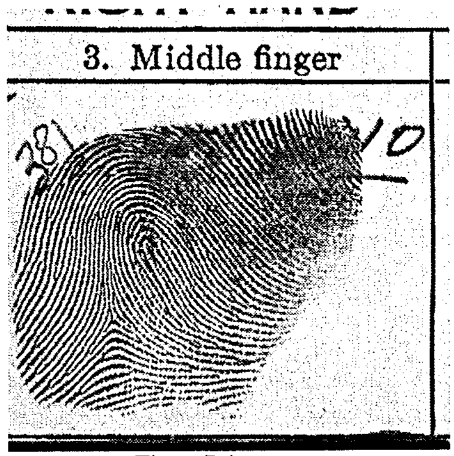

BrainMR_Top


NASA5

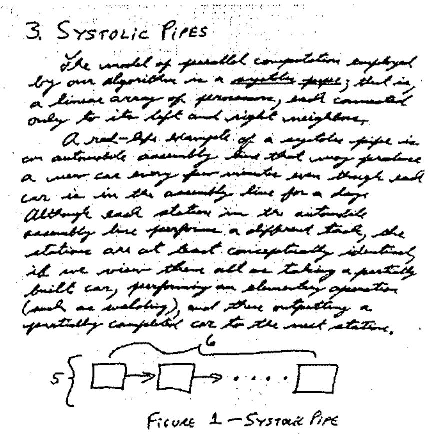

NASA6


LivingRoom

WomanHat



FingerPrint


3. Systolic Pipes

The model of parallel computation employed by our algorithm is a systolic pipe; that is, a linear array of processors, each connected only to its left and right neighbors.

A real-life example of a systolic pipe is an automobile assembly line that may produce a new car every few minutes even though each car is in the assembly line for a day. Although each station in the automobile assembly line performs a different task, the stations are at least conceptually identical if we view them all as taking a partially built car, performing an elementary operation (such as welding), and then outputting a partially completed car to the next station.

Figure 1 — Systolic Pipe

HandWriting

Electronic Democracy or Electronic Tranqualizer Where are we going on the Information Superhighway?

Barbara Simons*
Chair, U.S. Public Policy Committee of the ACM (USACM)

Technical and policy decisions can frequently have an unanticipated impact. For example, the post-WWII road building program resulted in the growth of the suburbs and the decline of the inner cities. A computing related example for which we do not yet know the long term impact is the U.S. government's decision to develop and promote escrowed key encryption — i.e. the Clipper Chip. Another recent example is the "Communications Decency Act of 1995" (S314), introduced by Sen. Exon, which would make all telecommunications service providers criminally liable for every message, file, or other content carried on their networks.

Some general issues include:

- Will the NII be primarily passive (movies and pizza on demand) or will it be strongly interactive, as is the case with the Internet?
- What type of privacy and security protections will be adopted? Some are necessary in order for electronic commerce to flourish, but what types will these be? What will be their side-effects?
- Will the government or other users be able to monitor communication contents? communications traffic?
- What is the meaning of "community standards" in pornography cases in which material is distributed nationally or even internationally? Who has legal jurisdiction?
- Will there be controls over the flow of information?
- How will intellectual property be protected?
- Will we bring the public schools on-line, and if so, how? At a time of government cutbacks, who will fund this initiative?
- Will citizen participation in democracy be increased?
- What is "universal access", and will it be provided?

Special interest groups are spending a lot of money in attempting to influence the answers to these questions. As the builders and early users of the net, we have considerable knowledge of the technical options and difficulties. What should be our role in providing input for the debate?

*IBM Santa Teresa Laboratory

Intellectual Property Issues on the Information Superhighway: Testimony of Barbara Simons, Chair USACM

This morning's meeting was chaired by Bruce Lehman, Ass't Sec'y of Commerce and Commissioner of Patents and Trademarks. Also present were Bruce McConnell, Chief, Information Policy/Technology Branch, OMB, Esther Dyson, President, EDventure Holdings, Inc, and Frances Preston, President and CEO, Broadcast Music Inc. Dyson and Preston are members of the NII Advisory Council.

Of the fourteen people who testified, I was the only one who represented the users. The testimony of Sandra Whisler, Ass't Director for Electronic Publishing, U. of Ca Press, and of Pieter Bolman, President, Academic Press, was thoughtful, reasoned, and balanced. But some of the other comments I heard were quite disturbing. One person seriously proposed that there be the ability to monitor the contents of everything that went over the net, so that violators of intellectual property could be apprehended. He seemed indifferent to my observation that such an approach not only violated notions of privacy but also would facilitate government monitoring of the citizenry. He also wanted to make bboard operators legally liable for everything that went over their bboard. Another person argued that an encryption scheme that was kept confidential was by its very nature more secure than one that was made public.

Although I suggested a metering method of billing for the net, I also pointed out in my comments that such a billing scheme implies major privacy concerns and, consequently, we would need additional legal protection to prevent vendors from selling information about individuals. While I was told that the NII committee is considering privacy issues, I did not have the feeling today that privacy protection was of primary concern to most of the folks who testified.

Barbara

P.S. William Ferguson, VP, Marketing and Sales, Semaphore Communications Corp, gave a very lively presentation in which he complained bitterly about how the gov't policy on the export of encryption software and hardware was constraining his and other U.S. businesses.

MEGA-PROJECT III of the INFORMATION INFRASTRUCTURE TASK FORCE ADVISORY COUNCIL and the SECURITY ISSUES FORUM of the INFORMATION INFRASTRUCTURE TASK FORCE

Comments of Dr. Barbara Simons
Chair, USACM, US Public Policy Committee, the Association for Computing Machinery
October 20, 1994

Thank you for the opportunity to speak with you today. As a representative of the computing profession, I particularly welcome the chance to discuss the development of the NII with the Advisory Committee.

I am here today on behalf of USACM, the Association for Computing Machinery's Committee on public policy. ACM is a non-profit educational and scientific society dedicated to the development and use of information technology, and to addressing the impact information technology has on the world's major social challenges. The 85,000 members of ACM are an outstanding resource for information, and we will be pleased to assist in any way we can. The ACM committee that I chair is particularly interested in policy and social issues involving network policy, including encryption, privacy, access issues, and computers in education. I have brought with me several documents, including an article I wrote listing questions about the NII and ACM and USACM statements on privacy, access, and the escrowed encryption standard. I have also brought a copy of our in-depth study of encryption policy in the U.S. entitled "Codes, Keys, and Conflicts: Issues in U.S. Crypto Policy". Incidentally, ACM is distributing this study free of charge on the net. I would like to say that it is a "best seller," but because it is available from several sites on the net, I have no idea of how many people have downloaded a copy.

I started using the net in the late '70s, while still a graduate student in computer science at U.C. Berkeley, and I could not function without it. I have used the net for such dissimilar activities as writing papers and running USACM. As an aside, even I have not met all the members of my committee. Nonetheless, we function very effectively using the Internet for our communications.

While I'm not quite an old-timer, I have been in the field long enough to witness the extraordinary computer-based revolution that has changed how we store and manipulate information. This revolution has made it possible for me to accomplish a great deal more than I could without this wonderful technology. But, this same revolution has also created some significant problems for industry. A digitally stored document or program can be disseminated for very little or no cost, either by shipping it over the net or by downloading it onto a floppy disk which can be given to someone else.

Consequently, we are confronted with the following questions:

1. Can we protect digitally stored intellectual property using:
    - technology?
    - financial disincentives?
    - new approaches?
    - the law?
2. What are the trade-offs of the various approaches?

It's not possible in the small amount of time that I have available to me to discuss any of these options in detail. So I shall state my views very briefly.

There are a variety of technical approaches for protecting intellectual property that one can contemplate. While it's impossible to develop a technique that is absolutely foolproof, people are currently working on technologies, using, for example, encryption, that are likely to discourage the vast majority of people from stealing intellectual property. An analogy can be made to the book publishing business. Photocopy machines can be used to copy books, but yet most people will choose to buy a book, rather than copy one that belongs to someone else.

The idea behind financial disincentives is that the cost of obtaining information should make it not worth one's while to copy and distribute protected material. An efficient and inexpensive automatic billing mechanism on the net could be used to process transactions that cost only a few cents or even a few dollars. If the costs are sufficiently low to obtain a copy legitimately, I'm unlikely to distribute my copy to my friends. There is material for which this approach would not be suitable, but cheap access could provide at least a partial solution that could be used in conjunction with other technologies.

New technologies frequently require new approaches. An interesting example is what happened in India to the Indian cable company Star. People started purchasing satellite dishes for reception of Star and then (illegally) selling the programming to apartment homes. Instead of attempting to arrest these small entrepreneurs, Star made their product free to everyone and used the large audience as a selling point for advertisers. Similar proposals have been made for the net, e.g. a bboard for dentists could be free to all and contain advertising that is aimed at dentists. Given the flexibility of the net, I would hope that any system that uses advertising to cover costs would allow the user the option of paying to not receive advertising.

I am not an expert on legal issues, and I do not intend to speak directly about them, or about the recently issued report on intellectual property rights. However, I am concerned that the law might be used as a blunt tool out of frustration from the lack of guarantees with other methods. We need to be very careful that we don't make laws that are routinely violated, both because of the selective enforcement aspects of such laws and because of the contempt for the law that is engendered by laws that are by and large unenforceable.

I have mentioned trade-offs to various approaches. We have to keep in mind that there are other important goals in addition to that of protecting intellectual property. In particular, there is the larger goal of promoting public access and use of the Internet. Like copyright generally, this goal also raises questions about crafting incentives that serve public interests.

Much of the development on the Internet has taken place without commercial incentives. This is not to suggest that commercial incentives should be discouraged. Rather, it is to remind the advisory committee that there are other important incentives for Internet users that should be preserved.

For example, there is within the user community a strong belief in sharing information and ideas as much as possible, except of course

where there are specific business restrictions. Many of the standards on the Internet evolved in an open, non-proprietary way. Even the popular program Mosaic has spread around the network without cost to the users.

The computer science community favors sharing because it promotes innovation, cooperation, and the development of good ideas. This is a spirit that you should be careful to preserve as the development of the national network moves forward.

There is also within the library community – a group I should add that now includes many computer science professionals – a similar commitment to the open exchange of information. Libraries work best when information is made freely available with as few obstacles to access and use as possible.

The growth of Internet sites with FTP capability, WAIS, Gopher, and Web access are recreating the library across the nation's electronic networks.

What are the benefits of the openness and the sharing that the computer science and library communities have promoted? Perhaps the most obvious is the growth of the Internet itself, which owes much of it development to the contributions of individuals working collaboratively without compensation to make communications technology more accessible and more useful.

There are larger social benefits as well. The spirit of openness has, for example, encouraged school children in inner city schools and rural parts of the country to explore new worlds, to learn new skills, and to make new friends. The recent grants from the NTIA should help further these efforts. Many members of the computer science community are very excited about these new projects.

Commercial applications of the Internet should be welcomed and encouraged. But so too should the continued growth of open and accessible networks that reach corners of our communities that might otherwise might be ignored.

Clearly, access to the Internet alone will not solve the many problems in our country. However, if we erect more barriers between communities, we will move further away from the goal of a technically literate, well trained workforce.

The Constitution speaks of copyright in the context of promoting "the Progress of Science and Useful Arts". The computer science profession has already made many contributions in this spirit. We hope that the IITF Advisory Committee will continue to encourage such efforts.

[end]

THE U.S. NATIONAL INFORMATION INFRASTRUCTURE - ACCESS ISSUES
Statement of USACM, the Public Policy Committee of ACM

The Association for Computing Machinery (ACM) endorses the creation of a National Information Infrastructure (NII) in the United States.

An NII that brings an open flow of information to U.S. citizens can improve economic well-being and can bring major advances in areas such as education, public health, public libraries, and a wide range of government and social services. As users of the precursors of the NII, ACM members are well aware of the benefits such a system can offer for business, education, communication, information, improved productivity, and quality of life.

USACM believes that such wide-reaching infrastructure must guarantee that the system be affordable and accessible for all. Access has several dimensions, most of which require public policy attention:

- Availability–An eventual NII must be geographically ubiquitous and accessible to everyone, both users and service providers.
- Protection of information rights–Privacy, property rights, public access rights, and freedom of speech will have to be protected. Lack of such protections will discourage public access and exchange of ideas.
- Affordability–Connection to a NII should be priced so that there can be universal access to a basic level of services. Also, access should be made available through public schools and public libraries, especially those in economically disadvantaged neighborhoods.
- Access to public services–The U.S. government will need to assure that applications with broad public benefit, such as interaction with government agencies and access to public data, are developed and made available.
- Lack of bias–Explicit efforts are needed to ensure that the NII addresses the entire spectrum of citizens and decreases the current cultural and gender gaps in technologically oriented services. All members of society should be encouraged to become information-technology literate.
- Ease of use–Access to the network and its basic services must be made so simple that even novices can use them and experts can work rapidly and effectively.

USACM believes that such an NII is technically feasible. That is not to say all problems are solved. Many of the technical issues are at the frontier of computer research and must receive proper attention. Many of the applications envisioned are large and complex, and will require the cooperation of much of the computer/communications industry, in areas that have posed substantial difficulties in the past.

We urge that the goals listed above be considered of primary importance in the research, design, and implementation of the NII, and that the broader public be included in the discussions among technical and political participants that will lead to decisions. It will not be easy to forge the necessary agreements among the many different voices to be heard, but it is a crucial part of shaping the information future.

USACM Position on the Escrowed Encryption Standard

The ACM study "Codes, Keys and Conflicts: Issues in U.S Crypto Policy" sets forth the complex technical and social issues underlying the current debate over widespread use of encryption. The importance of encryption, and the need for appropriate policies, will increase as networked communication grows. Security and privacy of electronic communications are vital to the development of national and international information infrastructures.

The Clipper Chip, or "Escrowed Encryption Standard" (EES) Initiative, raises fundamental policy issues that must be fully addressed and publicly debated. After reviewing the ACM study, which provides a balanced discussion of the issues, the U.S. Public Policy Committee of ACM (USACM) makes the following recommendations.

1. **The USACM supports the development of public policies and technical standards for communications security in open forums in which all stakeholders – government, industry, and the public – participate.** Because we are moving rapidly to open networks, a prerequisite for the success of those networks must be standards for which there is widespread consensus, including international acceptance. The USACM believes that communications security is too important to be left to secret processes and classified algorithms. We support the principles underlying the Computer Security Act of 1987, in which Congress expressed its preference for the development of open and unclassified security standards.

2. **The USACM recommends that any encryption standard adopted by the U.S. government not place U.S. manufacturers at a disadvantage in the global market or adversely affect technological development within the United States.** Few other nations are likely to adopt a standard that includes a classified algorithm and keys escrowed with the U.S. government.

3. **The USACM supports changes in the process of developing Federal Information Processing Standards (FIPS) employed by the National Institute of Standards and Technology.** This process is currently predicated on the use of such standards solely to support Federal procurement. Increasingly, the standards set through the FIPS process directly affect non-federal organizations and the public at large. In the case of the EES, the vast majority of comments solicited by NIST opposed the standard, but were openly ignored. The USACM recommends that the standards process be placed under the Administrative Procedures Act so that citizens may have the same opportunity to challenge government actions in the area of information processing standards as they do in other important aspects of Federal agency policy making.

4. **The USACM urges the Administration at this point to withdraw the Clipper Chip proposal and to begin an open and public review of encryption policy.** The escrowed encryption initiative raises vital issues of privacy, law enforcement, competitiveness and scientific innovation that must be openly discussed.

5. **The USACM reaffirms its support for privacy protection and urges the administration to encourage the development of technologies and institutional practices that will provide real privacy for future users of the National Information Infrastructure.**

March 22, 1995

Honorable James Exon
United States Senate
SH-528
Washington, DC 20510

Dear Senator Exon:

We are writing to you on behalf of the leading computing societies in the United States about the proposed Communication Decency Act. The memberships of our societies include scientists, engineers, and computing practitioners from every university, industrial research institution, government laboratory, and major computer firm in the United States.

We share your concern about the inappropriate and improper use of computer networks to send indecent material. However, we are deeply worried about the potential damage to our nation's communications infrastructure that seems likely to result from the Communication Decency Act as presently drafted. In particular, this legislation would impose unreasonable technical and financial burdens on the increasing number of institutions, large and small, that rely on the Internet for communication. We believe that these burdens will significantly harm the technological and communications opportunities now emerging from the Internet.

The growth of computer networks in the past two decades has been of enormous benefit to the entire country. It is in the national interest to continue encouragement of the technical innovation, economic growth, and world scientific leadership that our nation's computer networks have provided.

To allow a thorough exploration of the issues, we urge you to hold comprehensive hearings on the implications of the Communication Decency Act. We would be pleased to serve as a resource for you in this process, by providing analysis, expertise, and witnesses.

Many thanks for your consideration of our comments. We look forward to working with you.

Sincerely,

Barbara J. Grosz, President
American Association for Artificial Intelligence
Menlo Park, CA

Stuart H. Zweben, President
Association for Computing Machinery
New York, NY

Eric Roberts, President
Computer Professionals for Social Responsibility
Palo Alto, CA

Ronald Hoelzeman, President
IEEE Computer Society
Washington, DC

Margaret H. Wright, President
Society for Industrial and Applied Mathematics
Philadelphia, PA

Video And Image Semantics:

Advanced Tools For Telecommunications

Alex Pentland, Rosalind Picard, Glorianna Davenport, Ken Haase
The Media Laboratory
Massachusetts Institute of Technology

March 30, 1995

1 Introduction

Within the next decade, the majority of data carried over telecommunications links is likely to be visual material. The biggest problem in delivering video and image services is that the technology for organizing, searching, and presenting images is still in its infancy. Consequently the goal of the M.I.T. Media Laboratory's Advanced Tools for Telecommunications Project, funded by BT (British Telecom), is to develop tools for automatically understanding and using the semantics of video and image materials.

To support visual services, we must first be able to build multimedia databases quickly and cheaply. We must be able to extract and represent the content of the video clips and images sufficiently well that the computer can automatically select material that fulfills the needs of wide range of users and purposes. And finally, the computer must be able to automatically assemble this material into a coherent presentation.

Figure 1 shows the outlines of the system we are building. Video and image material is brought in over ISDN lines, parsed into keyframes, subjected to *semantics-preserving image compression*, and stored in an analogical database. This material can then be further annotated off-line. When users ask a question, the stored semantics and on-line similarity judgements are used to automatically assemble a multimedia presentation that can be sent out over the telecommunications network.

2 Semantics-Preserving Compression

Usually it is impossible to completely annotate a multimedia database. For instance, if we had 1,000 pictures of people, and we wanted to be able to compare people based on their appearance, we would have to enter almost 500,000 database entries!

For such comparison questions, it would be much better if the computer could "see" what is in the images, so that it could answer our questions by looking through the pictures. One problem with this approach is that images are just too large to efficiently store and search thousands of them.

To effectively search through images and video, you need to be able to express the *content* of the image in a very compact way. The ability to compress an image based on it's *semantic content* is is often called *semantic bandwidth compression*. We have extended this idea, and applied it to multimedia databases.

Our system functions by taking measurements of image features — brightness, edges, texture measures, etc. — and then using either the Karhunen-Loeve or Wold transforms to obtain

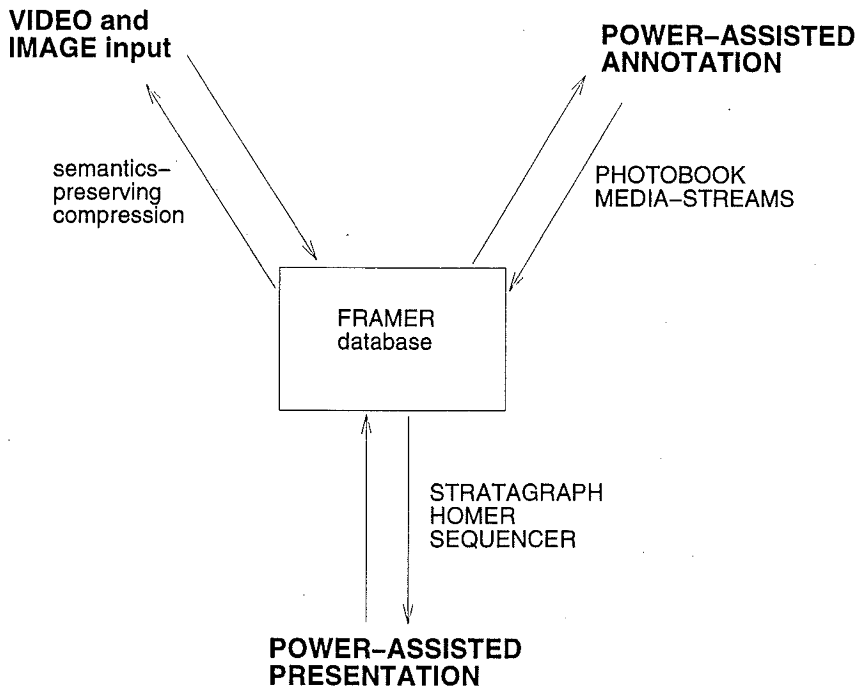


Figure 1: Overview of the system we are building: incoming video and imagery is subjected to semantics-preserving compression, and stored in an analogical database. Further annotations can be added off-line. When a user query is received, the stored semantics are used to automatically create an appropriate presentation

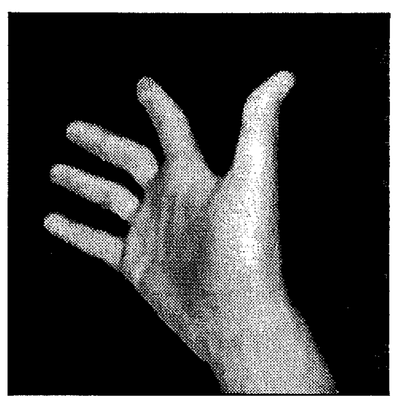
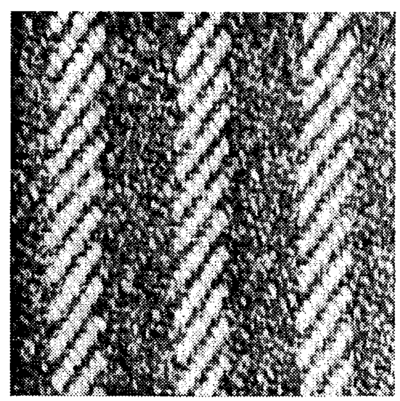
(a)  (b)  (c) 

Figure 2: Semantics-preserving compression. Shown here are three examples of images reconstructed from the coefficients used for database search. (a) 30 coefficients, (b) 100 coefficients, (c) 60 coefficients

(a)

(b)

Figure 3: Using motion and color information, we can separate foreground objects from background. This figure shows a system that extracts the outlines of people in view; a geometric analysis of the outline is then used to label position of head, hands, and feet. This system runs at 20 frames/second without special hardware.

a compact description of the set of images in terms of their most salient characteristics [2, 11]. The Karhunen-Loeve transform is used when the detailed relations between things are important, such when describing the geometry of a scene or a human face. The Wold transform is used when describing more textural properties, such as orientation, randomness, or periodicity.

In both cases the resulting representation of the image content can be searched directly, *without* decompression, to find objects and compare textures. This new representation technique, which we call *semantics-preserving compression*, can also provide an extremely compact code for image compression purposes. Some examples of semantics-preserving compression are shown in Figure 2.

An example of semantics-preserving compression applied to video is *keyframe extraction*. Editors and artists have long known that the semantic content of video can be accurately summarized by a series of appropriately-selected frames (images) taken from the video stream. These still-frame images are called *keyframes* and a sequence of them is called a *storyboard*.

Keyframes are images that are "characteristic" or "typical" of the video clip's content; we have found that good keyframes can be found by analysis of the camera and scene motion. For instance, good keyframes often occur in the middle of no-motion segments, and in the middle of segments where the camera is tracking a foreground object, as well as at the beginning and end of clips.

We can automatically extract such keyframes by computer analysis of the image motion in the video clip. By finding coherent subregions of motion in the video clip, we can automatically segment the scene into foreground, midground, and background, as illustrated in Figure 3. Then by comparison of foreground and background motions, we can automatically select useful keyframes [2].

3 Power-Assisted Browsing and Annotation

We have used this approach to create a browsing and database search tool called PHOTOBOOK [2]. This tool allows the user to browse large image databases quickly and efficiently, using both textual annotation information and by having the computer search the images directly based on their content. This allows people to search in a flexible and intuitive manner, using either

analogies, e.g., "show me this type of image," or visual similarities, e.g., "show me images that look like this." Figure 4 shows using PHOTOBOOK to find similar keyframes from a video database.

Using the Karhunen-Loeve approach, PHOTOBOOK has been shown to be 95% accurate at recognizing people, and over 99.9% accurate at verifing people's identity...accuracy figures that are competitive with fingerprints. Using the Wold model, PHOTOBOOK is also very effective at finding perceptually-similar textures in image databases. In testing of texture recognition, PHOTOBOOK has shown itself to be surprisingly accurate at mirroring people's *perceptual categories*, making it useful for finding semantically-similar textures.

4 Power-Assisted Video Presentations

Given a user query, the system uses "semantic templates" to search the database for entries that are relevant to answering the user's questions. However providing multimedia information is not like providing the latest cost figures from accounting. Each multimedia item shows only a small scene or action, so to provide information you have to string a series of images and video clips together so that they *tell a story*.

Because the material available for each query will be different, the machine must use similarity judgements (based on descriptions generated by semantics-preserving compression) together with analogical reasoning to decide what shots and stories best match the query. In our system this is accomplished using FRAMER, a persistent knowledge representation that uses analogical and similarity reasoning in addition to logical and set operations [6].

This allows the system to know which video clips are "right" for telling a particular story in the current context. Finally, the system assembles these clips using a story template, to create a video presentation that answers the user's question [7].

5 Conclusion

We have described a prototype system that is built on the idea of parsing video into semantically-meaningful chunks, and then encoding those chunks into a compact, easily-searchable representation that preserves the visual similarity relations. This *semantics-preserving compression* process can then be augmented with textual and analogical annotations. The result is a representation of the visual material that can be used to automatically assemble and efficiently edit multimedia presentations in response to user's needs.

References

[1] Haase, K., (1993) AI in Service and Support: Bridging the Gap, *Proceedings of the American Association for Artificial Intelligence*, 1993

[2] Pentland, A., Picard, R., and Sclaroff, S., (1994) Photobook:Tools For Content-Based Manipulation Of Image Databases *Storage and Retrieval of Image and Video Databases II SPIE PAPER 2185-05)* Feb 6-10, 1994, San Jose CA

[3] Picard, R., and Liu, F., (1994) A new Wold ordering for image similarity, *International Conference on Acoustic Signals and Signal Processing* March 1994, Adalaide, Austrailia. vol.5 page 129.

[4] Morgenroth, L., Davenport, G. (1994) "Let's See That Again: A Multiuse Video Database Project." Submitted to ACM Multimedia 1994, San Francisco, CA

Additional References:

[5] Davis, Marc. "Media Streams: An Iconic Visual Language for Video

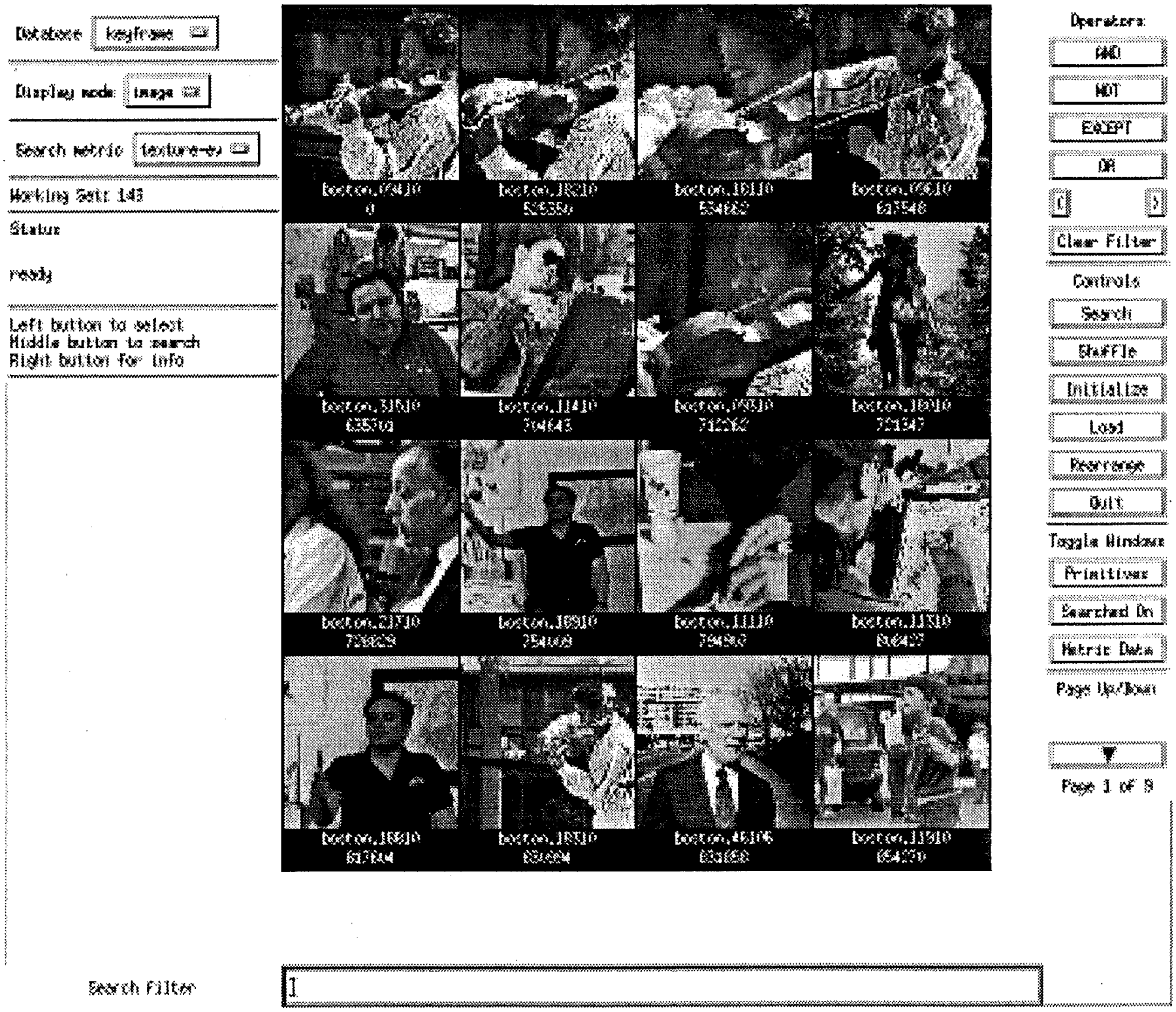


Figure 4: An example of a content-based image query: Are there any images similar to the image of the violin player shown at the top left? After searching a database of approximately 1,000 keyframes, the result is the series of images shown here. The images are ranked by similarity to the query image in terms of their visual content. Currently the system does surprisingly well...although usually there are some cases where it is difficult to understand the computer's similarity judgement.

Annotation." *Telektronikk* 4.93 (1993): 59-71. (Available on the WorldWideWeb at: http://www.nta.no/telektronikk/4.93.dir/Davis_M.html)

[6] Haase, K. (1993) "Framer: A Persistent Portable Representation Library." *Proceedings of the American Asso. for AI (AAAI-93).*

[7] MacKay, W. and Davenport, G. (1989) "Virtual Video Editing in Interactive Multimedia Applications." *Communication of the ACM* 32 (7 1989): 802-810.

[8] Morgenroth, L. (1992) "Homer: A Story Model Generator." B.S. Thesis, MIT, 1992.

[9] Pentland, A., Picard, R., Davenport, G., Welsh, R. (1993). "The BT/MIT Project on Advanced Image Tools for Telecommunications: An Overview." *ImageCom 2nd International Conference on Image Communications*, Bordeaux, France.

[10] Picard, R., (1992) "Random Field Texture Coding," *Society for Information Display International Symposium Digest*, Vol XXIII, May 1992, pages 685–688.

[11] Picard, R., and Liu, F., (1994) "A new Wold ordering for image similarity," *International Conference on Acoustic Signals and Signal Processing* March 1994, Adalaide, Austrailia. vol. 5, page 129.

[12] Picard, R., and Minka, T., (1994), "Vision Texture for Annotation" *ACM/Springer-Verlag Journal of Multimedia Systems, in press.*

[13] Sclaroff, S., and Pentland, A., (1994) "Modal Matching," *IEEE Trans. Pattern Analysis and Machine Vision, in press*

[14] Smith, A., Thomas, G. and Davenport, G. (1994). "The Stratification System: A Design Environment for Random Access Video." *ACM Workshop on Networking and Operating System Support for Digital Audio and Video* San Diego, CA.

[15] Turk, M., and Pentland, A., (1991) " Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, Vol. 3, No. 1, pp. 71-86.

NEW (OLD) MODELS FOR NETWORK-BASED LEARNING

Joseph V. Henderson, MD
Interactive Media Laboratory
Dartmouth Medical School
Hanover, NH 03769

Current, "Information Base" models of information access, particularly for the evolving Global Information Network and World-Wide Web, are limited. To be used effectively, users must have access to, and facility with, specialized systems, software and interfaces; there is little ability or effort to organize, contextualize, and develop coherent educational experiences.

Educators must work to take advantage of — and drive — the technological evolution of the Global Network. We propose the development and dissemination of educational programs that are rich in use of media and easy to use, adopting a more familiar, television-like "look and feel." Moreover, these programs should — without apologies — teach: they can provide coherent and comprehensive information about a domain, assist the learner in forming conceptual frameworks that provide a basis for other kinds of learning, and they can place what is learned in a rich, real-world, human context.

Joseph V. Henderson, MD, MA is Associate Professor of Community and Family Medicine, Associate Professor of Surgery, Dartmouth Medical School; Adjunct Associate Professor of Education and of Engineering, Dartmouth College. Dr. Henderson's main interest is in the use of computer, communications, and media technologies in education and in use of data visualization methods to provide access to large sets of multimedia information. Dr. Henderson is widely recognized for his qualities as a producer and designer of interactive media programs that have a very "human" feel, that stimulate the intellect and touch the heart.

The Web and Beyond: Agent-Based Publishing on the Internet

Brewster Kahle
President, WAIS Inc.

As managers and business people increase their use of the Internet as an information resource, new tools are emerging to better satisfy these user communities. This talk will address some of the technologies that facilitate research and alerting on the Internet.

The World Wide Web has paved the way for Fortune 500s, publishers, and government entities to make large databases available for attractive costs. Through compelling interfaces, many people have now found the Internet approachable. Like CD-ROM, the WWW has provided a mechanism to offer their database content in a digital form.

While gigabytes of content are rapidly becoming available, the tools are still primitive to filter or package this information.

Content alerting through "agent technology" has been identified as a promising direction for targeted delivery of packaged content for high-end users. Whereas the promise of an intelligent assistant that scours the net has not been delivered, tools are available and are becoming available to automate repetitive searches and to package the contents for quick browsing.

Some of the commercial tools that will be described that expand the Web for business users:

- Personal pages,
- Content Aggregators,
- Email Alerting, and
- Personal Digital Newspapers.

These tools promise to extend the Web past the "surfer" community to those who are time pressured and data hungry.

Publishing New Media For Higher Education

Edward Murphy
President, PWS Publishing Company

1. The Online Services Market

Market Size and Dynamics
Content Providers on the Net
Higher Education and Online Publishing
Marketing and Product Development Models
Where are the Likely Breakthroughs?

2. Multimedia Publishing

Educational Multimedia Market
Best Publishing Prospects and Why?
Role for Traditional Publishers
Partnership Models
Where are the Likely Breakthroughs?

3. The Curriculum as Customer

A Strategy for Publishing in the Information Age
Some Suggested Design Specifications
The Dissemination of Innovation
Barriers to Commercial Success

4. Information Technology in Higher Education

Infrastructure Spending Patterns
Classroom Use of Educational Technology
Distance Education: Examples from the Public and Private Sectors
Obstacles and Opportunities to Commercializing Educational Technologies

5. One Company's Approach

Strategy and Vision at PWS
The Chicken-Egg Dilemma: Where to Begin and How?
Organizational Issues: Changing Roles of our Authors and our Staff
Marketing Issues: Market-in vs. Product-out
Project-Specific Examples: PWS OnLine Calculus Modules, Developmental Mathematics, Design and Visualization

AsTeR –Towards Modality-Independent Electronic Documents

T. V. Raman
Digital Equipment Corporation
Cambridge Research Lab
One Kendall Square, Building 650
Cambridge, MA 02139
Voice-mail: `1 (617) 621-6637`
E-mail: ⟨`raman@crl.dec.com`⟩
WWW: http://www.research.digital.com/CRL/personal/raman/home.html

March 10, 1995

Abstract

The advent of electronic documents and the consequent creation of digital libraries —vast repositories of electronic information— has a profound impact on how we produce, organize, store, retrieve and consume information. All of these activities have been dictated to the present by the technologies used to share information; A change in the underlying technology, namely, the move from paper to electronic documents, offers a unique opportunity to revolutionize how information is archived and disseminated. This paper will focus on a specific aspect of the opportunities opened up by electronic publishing on the NII —the ability to present information in multiple modalities and thereby free it from any single presentation medium.

Traditional printed communication relies on a passive intermediary, paper,for the exchange of information between the author and reader. Ideas put down on paper come back to life only when perused by the reader.

Electronic publishing is mediated by a computer, an agent capable of processing the information. As a consequence, the ideas expressed by an author need no longer be bound to any single "display" form; nor does it require human intervention to translate the information from one displayed form to another. Electronic information can be processed and displayed in a manner best suited to each individual's needs. Thus, the advent of electronic documents makes information available in more than its visual form —electronic information can now be display-independent.

Traditionally, an electronic document has been viewed simply as digitally representing (or the means towards producing) the printed page.

Instead, we view the electronic document as the basic entity that represents information; we allow the information to be rendered in different ways —on paper, spoken, processed in different ways by a computer, etc. This change of viewpoint has allowed us to develop AsTeR (Audio System For Technical Readings) a computing system that *audio formats* electronic documents to produce *audio documents*. AsTeR can speak both literary texts and highly technical documents that contain complex mathematics. Moreover, the listener can ask to have parts of a document repeated in different ways: a document has many different spoken views.

The adequacy of the audio rendering depends on how well the electronic document captures the essential internal structure of the information. In this paper, we discuss capturing structure and give guidelines for authors to follow to ensure that their documents exhibit structure adequately.

In the context of the NII, the digital libraries of the future can be viewed as large information servers that allow multiple clients to access and display information in a format chosen by the user. By obviating the need to move physical media, e.g., printed paper or recorded tapes, the NII enables the ready dissemination of multimodal renderings of information.

1 Introduction

The advent of electronic documents and the consequent creation of digital libraries has a profound impact on how we produce, organize, store, retrieve and consume information. All of these activities have been dictated to the present by the technologies used to share information; A change in the underlying technology, namely, the move from paper to electronic documents, offers a unique opportunity to revolutionize how information is disseminated. The same electronic document can be printed, spoken, spoken in outline form over telephone lines or the Internet, processed automatically to extract certain kinds of information, and so on. This paper will focus on a specific aspect of the opportunities opened up by electronic publishing —the ability to present information in multiple modalities and thereby free it from any single presentation medium.

But for all this to be realized, the electronic document has to be considered as the key component, not the printed page. The electronic document is not the representation of the printed form; the printed form is *one* representation of the electronic document. This means that the electronic document has to be written to convey explicitly as much structure as possible –and details of any one presentation medium, such as the spacing between paragraphs on a printed page and length of time between speaking sentences, have to be abstracted out of the electronic encoding.

Information present in traditional printed documents comes to life only when it is perused by the human reader. Intelligent processing of such information therefore requires explicit human intervention. Intelligent processing —computing— can range from performing symbolic calculations on mathematical expressions occurring in a document, to translating the information to alterna-

tive display formats, e.g., audio, hypertext etc. To give a specific example, it requires a trained reader to make printed information available in spoken form[1]

Electronic communication, on the other hand, is mediated by an information processor rather than passive pieces of paper. This means that we can separate out the capture and storage of information from its presentation. Markup systems[2] like (LA)TEX capture the logical structure of a document along with its content. Rendering or presentation —the process of producing a "display"— can be viewed as applying a specific set of transformations to the abstract logical structure encapsulated by the encoding.

Typically, the structure is visually formatted to produce visual layout, a rendering attuned to the eye's ability to rapidly access different parts of a two-dimensional display. Thus, visual rendering projects the document logical structure on paper in a form that enables the reconstruction of the structure envisioned by the author.

Before getting into details of aural presentation, it will be useful to talk about the difference between printed and spoken documents. The passive printed document is processed by an active reader, who can view it in many different ways —read only section titles, skip a piece of mathematics, temporarily skip to a different page to read a referenced theorem, reread an interesting passage, and so on. Such active processing becomes even more flexible when the document appears on a computer screen, because hypertext and calculational capabilities can be used.

When it comes to audio, on the other hand, the *document* is the active player and the human the passive one. The speaker (perhaps on an audio cassette) actively reads in a relentlessly linear fashion, from beginning to end, and the listener simply listens, with little control over the process. Further, producing audio documents can be a laborious and time-consuming task —just ask organizations like Recordings For the Blind (RFB), who are engaged in producing such " talking books"[3].

ASTER –Audio Documents

ASTER (Audio System For Technical Readings) [Ram94] is a computing system that *audio formats* electronic documents to produce *audio documents.* Audio formatting produces renderings that are attuned to an auditory display. In its interactive mode, ASTER changes the active-passive relationship described

[1] Organizations like Recordings For the Blind (RFB) have been engaged in producing such "talking books" —an extremely laborious and time-consuming process.

[2] To most people, *markup* means an increase in the price of an article. Here, "markup" is a term from the publishing and printing business, where it means the instructions for the typesetter, written on a typescript or manuscript copy by an editor. Typesetting systems like (LA)TEX have these commands embedded in the electronic source. A *markup language* is a set of means (constructs) to express how text (i.e., that which is not markup) should be processed, or handled in other ways.

[3] An audio recording of the author's PhD thesis produced by ASTER is the first computer-generated talking book to be produced by RFB.

above by enabling *interactive listening*. The listener can browse the document structure and can obtain different *audio views* of (pieces of) the document.

The interested reader can experience an interactive demonstration[4] of the audio renderings produced by AsTeR on the WWW (available from the author's home page). It aptly brings out the power of the Internet in publishing multimedia documents; none of my journal publications come with online demonstrations. It also emphasizes the display-independent nature of electronic documents; both the audio formatted version and the visually laid out Postscript were generated from the *same* LaTeX source.

We envision digital libraries as repositories that serve information. Unlike libraries of today that store information in a single display format, the digital library of the future could potentially provide customized views of information. The rest of this paper will focus on the generation of multiple views of information objects. We emphasize that such multiple views can be multimodal, i.e., renderings may be visual, aural, and in the general case, a combination of both visual and aural views.

2 Representing Information

All information has structure, and any physical rendering of a document is a projection of this structure onto a particular medium, e.g., printed paper. A "rendering" of a document on some medium is best understood if it makes this logical structure readily apparent. For example, a visual rendering —onto a two-dimensional medium like paper— may use cues like boldface, different fonts, and indenting to help reveal structure. A visual rendering takes advantage of the eye's ability to rapidly access different parts of a two-dimensional display. An audio rendering has to use an entirely different set of cues to reveal structure.

Early in the development of AsTeR, we realized that the ability to render information in a variety of output modalities would be a direct function of the richness of the internal representation used to capture structure and content. Abstractly speaking, the high-level structure of a document is independent of any particular mode of display, and the internal representation should reflect this. As a first step in realizing AsTeR, therefore, we developed high-level models to represent document structure. For instance, the richness of the representation used by AsTeR completely frees the order in which subterms in an equation are rendered aurally from the order in which they would appear on paper. (See 4 for details.)

This section briefly outlines some of the representations used in AsTeR. Rendering this high-level representation is outlined in Section 4. Based on these ideas, we define a set of requirements in Section 5 that should prevent electronic encodings from being tied down to any single display form.

[4] This hypertext document presents a collection of math examples rendered in audio by AsTeR and in Postscript by LaTeX/DVIPS.

Ordering the Possible Representations

The amount of structural information that can be extracted from the electronic source depends entirely on how the logical structure is marked up. In the context of OCR-based document recognition, this is also a function of the quality of the visual rendering being recognized. In the case of both markup-based and OCR-based document recognition, the type of structure that can be extracted varies widely.

Intuitively, there is a hierarchy of document types (lattice) ordered by the amount of structural information captured, and the ease with which such structure can be recognized. The amount of structural information varies from plain paragraphs and sentences marked up with normal punctuation, all the way up to highly technical documents with footnotes, equations and references. The ease with which the structure can be extracted ranges from the bitmap on a low-resolution fax, through to a postscript[5] or PDF[6] file, on upward to a highly marked up LaTeX or SGML file. Given a document instance, the amount of structural information determines which of these logical structures we can extract. Given a plain ASCII document, structural information has to be inferred from the layout of the text, e.g., spacing, vertical alignment and centering. This is also true of pure visual layout encodings like PostScript and PDF. In the case of encodings in markup languages like (LA)TeX, much of the logical structure is explicitly present in the electronic source. Structure based document encoding systems like SGML provide the potential for extracting the richest possible logical structure, since they separate the layout process from the encoding of the document structure.

Document Models in AsTeR

The recognizer used in AsTeR captures logical structure present in documents encoded in the TeX family of languages. An important feature of this recognizer is that it works on the entire gamut of encodings, ranging from plain ASCII documents, i.e., no explicit markup, up to documents containing completely unambiguous encodings of the logical structure.

The basic document model used in AsTeR is the attributed tree. Each hierarchical level of the document is modeled as a node in this tree. Each node can have content, children and attributes. Using object-oriented terminology, each different kind of node of the tree is called an *object*. Thus, "chapter", "section", "paragraph", and "sentence" are all objects. If a document contained five sections, its representation in AsTeR would have five instances of object "section". This object-oriented terminology is used because AsTeR actually uses CLOS objects in this fashion. The use of an object-oriented language was instrumental in allowing us to develop and implement the ideas in AsTeR incrementally and effectively.

[5] PostScript is a registered trademark of Adobe Systems Incorporated.
[6] PDF, Portable Document Format, is a registered trademark of Adobe Systems.

$$\begin{array}{ccc} \text{left-superscript} & \text{accent} & \text{superscript} \\ \nwarrow & \uparrow & \nearrow \\ & \textbf{math object} & \\ \swarrow & \downarrow & \searrow \\ \text{left-subscript} & \text{underbar} & \text{subscript} \end{array}$$

Figure 1: A math object with attributes. Each of the attributes themselves contain math objects.

This attributed tree structure is augmented to represent mathematical content; we call this augmented representation the *quasi-prefix form*, (see figure Figure 1 on page 6). Expressions that are completely unambiguous, e.g., $x + y$, are captured in their prefix form. In addition to linearizing the underlying tree structure, mathematical notation uses ***visual attributes*** such as superscripts and subscripts, whose interpretation is context-dependent. We extend the prefix form to capture such visual attributes —hence the name *quasi*-prefix.

A key feature of the quasi-prefix form is that it delays the assignment of semantic interpretation to instances of ambiguous written mathematics. At the same time, it is sufficiently rich to permit renderings that are independent of the order in which the written symbols would appear on paper. Linear renderings with the rendering-order hard-coded into the system can be produced with a simpler representation, e.g., a linear list, or even the TEX encoding itself. This was shown by TEXTALK, a string-substitution based program that directly transformed TEX source to produce spoken renderings [Ram91, Ram92].

As an example, assume that `\kronecker`[7] is defined as an infix binary operator. Given the expression

$$a \otimes b$$

encoded as

```
a\kronecker  B
```

we can represent it in the quasi-prefix form by a tree whose root is object *kronecker*, and write rendering rules for object *kronecker* that produce either "a kronecker product b", or "kronecker product of a and b". The former rendering can be produced by TEXTALK as well, but a simpler list-like representation restricts the system to this one form of rendering.

In producing printed output, one view is sufficient; once the information has been presented visually, a person reading the material can access it in any desired order. But even with visual rendering, different views may be desired. For example, one may wish a view that gives only the table of contents of a paper. Or, for a document that presents an algorithm, one view could give the whole presentation and a second view could present only the overview of the algorithm. See [Lam93] for a discussion on the hierarchical presentation of

[7] TEX does not provide this operator, and it will have to be defined as a macro. We describe how ASTER is extended to handle such macros in Section 3.

proofs. The linearity of audio makes it essential that AsTeR have the ability to present multiple views. Lack of this feature is one of the major shortcomings of books on tape, where the listener is restricted to the one view presented by the person speaking the text. AsTeR allows the listener to explore the material the same as a person perusing printed material, and thereby enables *active* listening.

3 Extending Document Logical Structure

A flexible markup system needs to be extensible. This is because it is impossible to enumerate all possible logical structures that different authors might wish to use. This is perhaps one of the biggest shortcomings with SGML, where extending the logical structure requires modifying the Document Type Definition (DTD), a non-trivial task. On the other hand, one of the most powerful aspects of (LA)TeX is its extensibility. It is possible for an author to define the few additional logical constructs needed by a particular document instance using the TeX macro facility, [Knu84, Knu86].

Macros permit the author to abstract away layout details when writing the document. To give an example, the command `\kronecker` is not present in (LA)TeX. An author can extend (LA)TeX by defining

```
\def\kronecker{\raisebox{1pt}{\:\otimes\:}}
```

and then write

```
$ A \kronecker B$
```

The definition for `\kronecker` has extended the markup language, and consequently, the logical structure that can be expressed. LaTeX [Lam86] is itself a good example of how TeX macros can be used to implement a language for encoding document structure. The presence of user-defined macros in documents presents both a challenge and an opportunity for a system like AsTeR.

The Challenge

In general, TeX macros can perform any arbitrary computation permitted by TeX, a Turing-complete language. Hence, it is impossible to directly translate the macro expansion into an audio rendering. The TeX primitives are visual layout operators, and translating a TeX macro directly into an audio rendering rule would imply a one-to-one mapping between the visual and audio rendering. As explained in Section 1, visual renderings are attuned to a two-dimensional display, and audio renderings need to be attuned to an auditory display. Further, expanding a TeX macro loses structural information; when all macros in a document have been expanded, only the visual layout remains.

The Opportunity

A proper choice of macro definitions can encode semantic information about the visual layout being used. For instance, an author wishing to use the notation

$(\frac{P}{Q})$ to denote the Legendre symbol could define a new command `\legendre` that produces the desired layout. In the absence of this facility, e.g., when using an SGML Math DTD that does not know about Legendre symbols, the author would be restricted to encoding $(\frac{P}{Q})$ in terms of the basic operators provided, in this case, the operator that stacks its arguments. Though both encodings would produce the desired visual layout on paper, the encoding using the `\legendre` macro has the advantage that an application can interpret the use of the notation as meaning a Legendre symbol.

Representing Extended Logical Structure

The first step in solving this problem is to represent instances of user-defined macros in our high-level document model. Producing audio renderings (or renderings in other modalities) of such instances is then equivalent to rendering any other object present in the model.

We model macro definitions as introducing new object types. Thus, defining `\legendre` is equivalent to adding object *legendre* to the set of objects present in the document model. A macro definition in (LA)TEX has two parts; the first part declares the macro and its number of arguments; the second part specifies how instances of this macro call are to be displayed.

Translating this to the object-oriented model, the first part of the macro introduces a new object type; the second part is a visual rendering rule for instances of this object. We extend ASTER to handle user-defined macros according to this framework. Hence, for every user-defined macro, the document model is extended by adjoining a new object type, and calls to that user-defined macro are represented by instances of this new object type.

4 Rendering Information Objects

ASTER renders information by applying *rendering rules* to the internal representation described in Section 2 and Section 3. The system of rendering rules used in ASTER and the language in which they are written (AFL —Audio Formatting Language) are described in detail in [Ram94]. In a sense, AFL is to audio formatting as Postscript is to visual formatting, although AFL is a much smaller language.

Here, we show a small example of such a rendering rule for a user-defined macro. In the following, we use CLOS generic function read-aloud. For the present, let us assume that function read-aloud executes the necessary actions to render its argument.

After extending ASTER to process the (LA)TEX macro `\inference`, which is defined as

```
\newcommand\inference[2]{\frac{#1}{#2}}
```

to render instances of calls to `\inference`, we can define

```
(defmethod read-aloud((inference inference))
  "Sample rendering  for object inference."
  (read-aloud (argument 1 inference))
  (read-aloud "implies")
  (read-aloud (argument 2 inference)))
```

Given $\frac{A}{B}$, this produces "A implies B".

If we wished to produce a rendering that inverts the order in which the arguments to macro `\inference` are rendered, we would define:

```
(defmethod read-aloud((inference inference))
  "Renders inference with arguments reversed."
  (read-aloud "We know that ")
  (read-aloud (argument 2 inference))
  (read-aloud "because")
  (read-aloud (argument 1 inference)))
```

which produces "We know B because A".

Switching between these two rendering rules has the effect of inverting a proof-tree! AsTeR makes it easy to write several rendering rules for the same object. AsTeR also allows rendering rules to be partitioned into rendering *styles*. In an interactive session with AsTeR, switching between rendering styles (a collection of rendering rules for different objects) and invoking individual rendering rules can be done with a few keystrokes, making it easy for a listener to obtain many different views of a document.

AsTeR derives its power from representing document content as objects and by allowing multiple user-defined rendering rules for individual object types. These rules can cause any number of audio events (ranging from speaking a simple phrase, to playing a digitized sound). The pitch of the voice, the physical head-size of the virtual speaker, the volume, and many other parameters can be changed by rendering rules, making it easy to create sound cues to help display structure.

To give an example of this, the logo for AsTeR is

and is produced by (LA)TEX macro `\asterlogo`. After appropriately extending AsTeR to recognize this macro, we can define an audio rendering rule that produces a bark when rendering instances of this macro. Thus, the same piece of markup `\asterlogo` produces the picture of Aster[8] when rendered visually, and an appropriate sound[9] when rendered aurally.

[8] Aster is my guide-dog.

[9] The bark is that of a generic dog, Aster is too well trained to bark, and could not therefore be recorded.

This feature was exploited to advantage when producing the audio formatted version of the author's thesis. The dedication page of the thesis contains a large picture of Aster, and the audio formatted version contains a verbal description of the picture, accompanied by the sound of Aster panting in the background. You can listen to this example on the WWW —visit the AsTeR home page[10] and click on the picture of Aster.

Several ideas come together to make all this possible. First, logical structure is of paramount importance —not its display on any one particular medium. The more a document makes structure explicit, the better the document can be displayed on several different mediums.

Next, the use of (LA)TEX macros to encode structure makes it possible to have a system like ASTER, in which the internal structure can be extended to fit a document. This allows the encoding of the structure in a flexible, uniform, and consistent representation such as an attributed tree, with the addition of the quasi-prefix form for dealing with mathematics.

Finally, providing different rendering rules and styles and a flexible way to switch among them makes it possible to obtain multiple views of a document in an interactive fashion.

5 Conclusion

We conclude this paper with a few guidelines for encoding document content in a display-independent manner. Electronic encodings that adhere to these guidelines will enable multiple uses of the same electronic source. Though the notion of archiving information in its richest possible form is itself not new, we note that such ideas have been exclusively motivated in the past by the need to display information visually. The richest representation for the specific problem of being able to accurately reproduce the visual appearance of information is not necessarily appropriate for *computing* on the information. Visual presentations, as pointed out earlier, are optimized for human consumption, and therefore necessitate explicit human intervention in performing *intelligent* manipulation of the content.

Our work brings a fresh perspective to this issue by addressing the problem of aurally rendering complex information. It points out that the visual presentation that we are all familiar with, e.g., the printed version of this paper, is just one possible view of the information content, not the information itself. This insight leads naturally to the approach used in ASTER, namely the development of high-level information representation and the rendering of such representations in different modalities.

To ensure a multiplicity of uses, the digital library should archive information in its richest form. Such encodings should be capable of producing high-quality renderings in the various output modalities, e.g., a well-formatted PostScript or PDF file containing high-resolution fonts, audio renderings that exploit the various features of an auditory display, etc. A digital library may choose to archive

[10] *URL* http://www.research.digital.com/CRL/personal/raman/aster/aster-toplevel.html

one or more of the "display" forms in addition to the high-level encodings as a means of optimizing information delivery. However, archiving information in any of these "display" forms is equivalent to archiving information on printed paper. Hence, such "display" representations should not be viewed as a replacement for the high-level encoding.

Retaining the high-level encoding that generates the various renderings will facilitate:

- Linking multiple views of the information.
- Producing additional views of the information.
- Searching the information.
- Computing on the information in as yet unforeseen ways.

The Chicago Journal of Theoretical Computer Science is an online journal to be published in LaTeX (*URL* http://cs-www.uchicago.edu/publications/cjtcs/) and fulfills these ideals. The markup recommended to authors has been carefully designed to abstract out all layout details by the Managing Editor, Prof. Mike O'Donnell, and we hope to aurally render articles from the journal using AsTeR. See [O'D92, O'D93] for a description of the work leading up to this project.

Unambiguous Encodings

The same visual layout may be used to display disparate concepts. Encoding instances of such ambiguous notation by using well-designed markup abstracts out the layout details from the document encoding, and allows an information agent to identify the different concepts correctly. We illustrate this with a concrete (LA)TeX example.

The visual layout of stacking one mathematical object above another, separated by a horizontal line (horizontal rule), could be used in several contexts.

- **Fraction:** $\frac{1+\sqrt{5}}{2}$
- **Inference rule:** $\frac{X=Y,\ Y=Z}{X=Z}$

Using the encoding `\frac{object-1}{object-2}` in both cases makes it impossible to disambiguate between the different interpretations. When the same layout is used to denote different concepts, these should be marked up distinctly.

For instance, in LaTeX, the author could extend the markup language by defining two new macros:

1. `\def{\fraction}[2]{\frac{#1}{#2}}`.
2. `\def{\inference}[2]{\frac{#1}{#2}}`.

Though stated in terms of (LA)TEX, the above requirement can be generalized to any encoding system. It merely states that objects that are semantically distinct but share a common visual representation should have distinct electronic encodings. This is essential in ensuring that such objects can be presented in other modalities, where they may not necessarily share the same displayed representation. More generally, such distinct encodings are also essential if we are to compute on the content encapsulated by the encoding.

Summary

- Avoid using any display-specific format as the principal form of archiving electronic information, e.g., a scanned bitmap image, a PostScript or PDF file (visual rendering) or a digitized recording (audio).
- Avoid use of explicit visual layout in the electronic encoding. For instance, avoid use of `\vskip` in (LA)TEX documents.
- Use distinct markup to encode semantically distinct objects even if they have the same visual layout.
- Use an encoding system that is extensible by the author; this will ensure that the maximum amount of semantic information is captured at the encoding stage. This minimizes the amount of guesswork that has to be done later.

Electronic document encodings have not always followed these rules, since the markup was viewed purely as a means of producing the visual rendering. Our work points out that the same encoding can be put to multiple uses; it is therefore important to apply principles of good software design and reuse to document encodings as well.

To draw an analogy, we do not currently throw away the program source code once we have successfully compiled it into a running executable; equivalently, it is important to retain the high-level document encodings that produce the final display form in which information is disseminated.

References

[Knu84] Donald E. Knuth. *The TEXbook.* Addison-Wesley, Reading, Massachusetts, 1984.

[Knu86] Donald E. Knuth. *TEX The Program.* Addison-Wesley, Reading, Mass., 1986.

[Lam86] Leslie Lamport. *LATEX: A Document Preparation System.* Addison-Wesley, Reading, Mass., 1986.

[Lam93] Leslie Lamport. How to write a proof. (94), February 1993. To appear in *American Mathematical Monthly.*

[O'D92] Mike O'Donnell. Electronic journals—scholarly invariants in a changing medium. *Conference on Academic and Professional Journals in the Twentieth Century*, April 1992. Presented by author as discussant for session on "The Future of Journals".

[O'D93] Mike O'Donnell. Issues involved in publishing an electronic journal. *Seminars on Academic Computing*, August 1993. Revised version of paper appeared as University of Chicago Department of Computer Science Technical Report 93-11, July 1993.

[Ram91] T. V. Raman. TEXTALK. *TUGboat*, 12:178, March 1991.

[Ram92] T. V. Raman. An audio view of (LA)TEX documents. *Proceedings of the TEX Users Group*, 13:372–379, July 1992.

[Ram94] T. V. Raman. *Audio System for Technical Readings.* PhD thesis, Cornell University, May 1994. *URL* http://www.research.digital.com/CRL /personal/raman/raman.html.

World Wide Web: The Consortium and Plans for the Future

Tim Berners-Lee
Director, W3 Consortium

Summary

The goal of the World Wide Web Consortium (W3C) is to ensure the evolution of the World Wide Web (W3) protocols into a true information infrastructure in such a fashion that smooth transitions will be assured both now and in the future.

Toward this goal, the teams at MIT and INRIA will develop, support, test, disseminate W3 protocols and reference implementations of such protocols and be a vendor-neutral convenor of the community developing W3 products. In this latter role, the team will act as a coordinator for W3 development to ensure maximum possible standardization and interoperability.

Currently, there are approximately 40 members of the Consortium, including companies like America Online, AT&T, Netscape, Sun and NTT Japan. The members form the Advisory Committee which consults with the W3C Director to more fully define the priorities and activities of the Consortium. Therefore, the work described herein is a starting point as to what W3C perceives to be important to the evolution of the Web. The technical part of the work falls into the following broad categories:

- Automatability: the ability to replace frequently used manual procedures with automated ones.
- Extensibility: the ability for new ideas, concepts, operations and object types to be incorporated into the Web incrementally and with back-compatibility
- Scalability, Efficiency, and Robustness: the properties which maintain in the operation of the Web in the face of changes in technology and in dramatic growth in size and usage
- Incorporation of Privacy: Web mechanisms for adding the privacy, data integrity and authentication required for commercial or confidential use.

Current activities of the Consortium includes specifically work in the following areas:

- Security and Payment systems
- Protocols for replication and caching
- Collaboration, Knowledge Representation, and Automatiability
- HTML levels 3 and 4
- The use of the web for general SGML applications
- Style sheet definition
- Content labelling and legal issues

For more information:

`http://www.w3.org/`
about The World Wide Web

`http://www.w3.org/hypertext/WWW/Consortium/`
for specific information about the W3 Consortium.

PAPER PRESENTATIONS

Content-based Image Retrieval: Color and Edges

Robert S. Gray*
Department of Computer Science
Dartmouth College

Abstract

One of the tools that will be essential for future electronic publishing is a powerful image retrieval system. The author should be able to search an image database for images that convey the desired information or mood; a reader should be able to search a corpus of published work for images that are relevant to his or her needs. Most commercial image retrieval systems associate keywords or text with each image and require the user to enter a keyword or textual description of the desired image. This text-based approach has numerous drawbacks – associating keywords or text with each image is a tedious task; some image features may not be mentioned in the textual description; some features are "nearly impossible to describe with text"; and some features can be described in widely different ways [Na93a]. In an effort to overcome these problems and improve retrieval performance, researchers have focused more and more on *content-based* image retrieval in which retrieval is accomplished by comparing image contents directly rather than textual descriptions of the image contents. Some content-based systems require specific knowledge about the domain from which the images are taken. Such domain knowledge is tedious to construct and maintain – especially for an end user – and should be avoided *if it is possible to achieve good retrieval performance through domain-independent techniques.* Many such techniques have been proposed. Most retrieve images on the basis of simple features such as color, shape, texture and edges. In this paper we describe a content-based system that retrieves images on the basis of their color distributions and edge characteristics. The system uses two retrieval techniques that have been described in the literature – i.e. *histogram intersection* to compare color distributions and *sketch comparison* to compare edge characteristics. The performance of the system is evaluated and various extensions to the existing techniques are proposed.

1 Introduction

One of the tools that will be essential for future electronic publishing is a powerful image retrieval system. The author should be able to search an image database for images that convey the desired information or mood; the reader should be able to search a corpus of published work for images that are relevant to his or her needs. Most commercial image retrieval systems associate keywords or text with each image and require the user to enter a keyword or textual description of the desired image. Standard text retrieval techniques are used to identify the relevant images in the corpus. For example, the Kodak Picture Exchange (KPX) – an on-line database that contains over a hundred thousand photographs from seventeen photo houses – has a keyword description for each photograph that specifies the objects in the photograph and the layout of the photograph [Ben94]. The user searches the database with Boolean keyword queries. Unfortunately the text-based approach to image retrieval has numerous drawbacks [Na93a]. Associating keywords or text with each image is a tedious and time-consuming task since it must be done manually or at best semi-automatically; image processing technology is not advanced enough to allow the automatic construction of textual image descriptions except in well-defined and tightly focused domains. Some image features may not be mentioned in the textual description due to design decision or indexer error; these image features

*Partially supported by AFOSR contract F49620-93-1-0266 and AFOSR/DARPA 89-0536

do not exist from the standpoint of the retrieval system and any query that mentions them will fail. Some features are "nearly impossible to describe with text" [Na93a]; for example many textures and shapes defy easy description. Finally different indexers – or even the same indexer – may describe the same feature with different terms or different features with the same term; these are the standard text retrieval problems of synonymy and polysemy.

In an effort to overcome the problems of the text-based approach and improve retrieval performance, researchers have focused more and more on *content-based* image retrieval in which retrieval is accomplished by comparing image contents directly rather than textual descriptions of the image contents. Some content-based systems require specific knowledge about the domain from which the images are taken. The most successful example is the Condor system which was developed at the Artificial Intelligence Center of SRI International. Condor was designed to perform object *recognition* but can be adapted easily to image retrieval. Condor is a production system that automatically recognizes objects in an outdoor scene – e.g. trees with a mountain in the background – and then constructs a 3-D model of the scene [SF91]. Production rules hypothesize that an object is in the scene on the basis of known objects and the results of primitive image operations. The domain knowledge represented in the rules is essential for accurate recognition [SF91]. However such knowledge becomes undesirable if Condor is used for image retrieval. The rules are tedious to construct and maintain – especially for an end user – and must be changed for every domain. The number of rules becomes unmanageable in large, heterogeneous databases. Finally the rules may not detect the particular object or feature that interests the current user or may not agree with the current user's concept of an object. Condor is an extreme example since it was designed for recognition rather than retrieval. However similar comments hold in most cases. Domain knowledge is an investment that should be avoided *if good retrieval performance can be achieved with straightforward, domain-independent techniques.*

Many such techniques have been proposed. Most retrieve images on the basis of simple features such as color, shape, texture and edges. It is hoped that these domain-independent techniques will form the basis for powerful "*query by example*" retrieval systems. For example, the user might provide a sample image and request similar images, *draw* a simple sketch of an object and request images that contain the object, or select a set of colors and request images that contain those colors. In this paper we describe a simple query-by-example system that retrieves images on the basis of their color distributions and edge characteristics. The system is fully automatic as no manual intervention is required before or during the indexing process. The system does not develop any novel retrieval techniques but instead uses existing techniques that have been described in the literature – i.e *histogram intersection* [SB91, Swa93] is used to compare color distributions and *sketch comparison* [HK92, KKOH92] is used to compare edge characteristics. It is hoped that the system will highlight potential avenues of research and serve as a testbed for future work. To this end, the performance of the system is evaluated and various extensions to the existing retrieval techniques are proposed. The next section describes the implementation of the system. The remaining sections discuss the weaknesses of the current implementation and methods for addressing those weaknesses.

2 Implementation

The system is implemented as four modules – edge extraction, color extraction, query processing and user interface. The color and edge extraction modules construct a set of histograms and an edge map for each image. No manual intervention is required during this process. The query processing module uses histogram intersection [SB91, Swa93] to compare histograms and sketch comparison [KKOH92, HK92] to compare edge maps. The user interface provides a graphical front end. The four modules are shown in figure 1 and described below.

2.1 Edge extraction

The *edge extraction* module originally used the edge detection algorithm from [KKOH92, HK92] which identifies the edges in an RGB

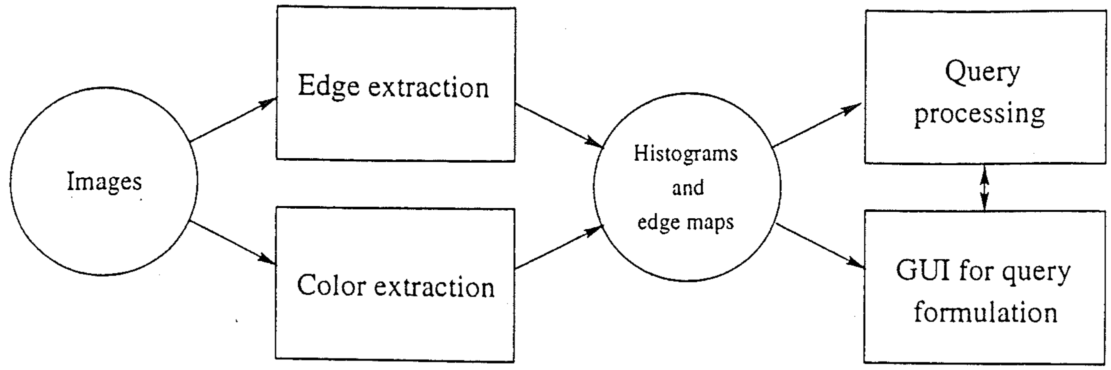


Figure 1: The four components of the image retrieval system

image that are *clearly perceptive* to a human viewer. First the RGB image is reduced to thumbnail size and median filtered. Four gradients – one for each major orientation – are calculated for each pixel in the thumbnail. Each gradient is scaled by the reciprocal of the local intensity power $| I_{ij} |$ which is defined as

$$| I_{ij} | = \left\{ \frac{1}{9} \sum_{r=i-1}^{i+1} \sum_{s=j-1}^{j+1} \overline{p}_{rs}^2 \right\}^{\frac{1}{2}}$$

where (i,j) is the pixel for which we are calculating the gradient and $\overline{p}_{rs}$ is the vector of RGB intensity values at pixel (r,s). This scaling factor is a simple application of the Weber-Fechner law to the RGB color space. The Weber-Fechner law states that the "contrast sensitivity of the human eye is proportional to the log-scale of the intensity value" [HK92].

The overall gradient for each pixel is taken to be whichever of the four gradients has the maximum absolute value. These overall gradients are used to identify the edge pixels. First the algorithm calculates the average and standard deviation of the gradient magnitudes over the entire thumbnail image. All pixels for which the gradient magnitude is greater than the average plus one standard deviation are marked as *global edge candidates.* Then the algorithm filters the set of global edge candidates by examining the local context of each candidate. It calculates the average and standard deviation of the gradient magnitudes over a small window centered on the global edge candidate and keeps the candidate only if its gradient magnitude is greater than the local average plus one local standard deviation. Finally an edge map is constructed in which a pixel is on if it is one of the final edge candidates and off otherwise. The goal of this technique is to extract only those edges that are *clearly perceptive* within their local section of the image and within the image as a whole. The resulting edge map should be generally similar to a human's impression of the image [HK92].

Figure 2: A sample image from the image database

The algorithm was applied to a test collection of forty-eight outdoor scenes that are sold as part of the Microsoft Scenes screen saver. The results were poor as the algorithm missed many clearly perceptive edges. This suggested that scaling by the local intensity power was an insufficient transformation of the RGB color space and motivated a switch from the RGB color space to the CIE-LUV color space. The CIE-LUV color space has the advantage that the distance between two points in the color space is approximately proportionally to the perceptual distance between the corresponding colors (as expressed

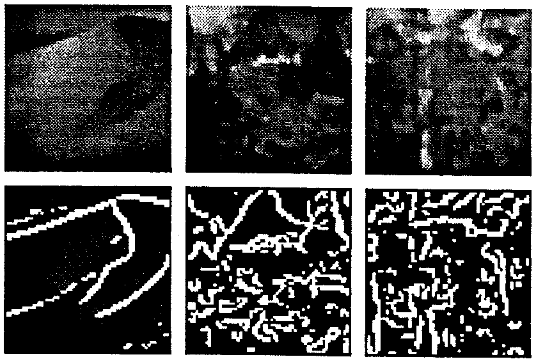


Figure 3: The performance of the edge detection algorithm

by human viewers). Our revised edge detection algorithm first converts the RGB image to a CIE-LUV image and then uses the detection algorithm as described above except that the gradients are no longer scaled by the local intensity power. Details of the RGB-to-CIE-LUV conversion can be found in [FVDFH91].

The revised algorithm performed much better. However edge map quality remained poor for a significant minority of the images. Figure 3 shows a few of the test images and their corresponding edge maps. Evaluation of edge map quality is necessarily subjective, but broadly speaking 45 percent of the maps contained every important edge (plus a small amount of noise); another 45 percent contained some important edges and some unimportant edges; and the remaining 10 percent contained no important edges. The main problem with the images that fall into the latter two categories is that many of their important edges are *clearly perceptive* only when texture or domain knowledge is considered. The edge detection algorithm considers color only.

We constructed 64 x 64 edge maps and used a 3 x 3 window when median filtering and a 7 x 7 window when filtering the global edge candidates. These values were used in [HK92]. Modified values did not produce significant improvements in retrieval performance or edge map quality.

2.2 Color extraction

The *color extraction* module divides each image into non-overlapping subareas as in [CLP94] and then constructs a three-axis CIE-LUV histogram for each subarea and for the overall image. We divided the image into 4 subareas – one for each quadrant of the image – and had 8 buckets along each color axis. These values provide reasonable retrieval performance.

2.3 Query processing

The *query processing* module accepts a single set of histograms and a single edge map as a query (a *set* of histograms consists of one histogram for each subarea and one histogram for the overall image). Then the module computes color and edge similarity scores for each image and takes the weighted average of the two scores to get an overall similarity score. The images are presented to the user in order of decreasing similarity.

The module uses histogram intersection [SB91, Swa93] to compare the query histograms against each set of image histograms. The system either compares just the overall histograms *or* compares each pair of subarea histograms and then takes a weighted average of the subarea similarity scores. Histogram intersection computes the similarity between an image histogram I and a query histogram Q as

$$S(I,Q) = \frac{\sum_{j=1}^{n} min(I_j, Q_j)}{\sum_{j=1}^{n} Q_j}$$

where n is the number of buckets in the histograms, I_j is the number of pixels in bucket j of the image histogram and Q_j is the number of pixels in bucket j of the query histogram. Histogram intersection was originally developed to identify which images *contain* an object . Thus the purpose of the *min* in the similarity measure is to filter out the background pixels and leave only those pixels that might belong to the object. Histogram intersection has been shown to be insensitive to "change in image resolution, histogram size, occlusion, depth and viewpoint" [Swa93] and has provided excellent performance when finding images that contain a given object [SB91]. These results should hold when the technique is used to determine the similarity between two equal sized images.

The module uses a slightly modified version of sketch comparison [HK92, KKOH92] to

compare the query edge map against each image edge map. First the query edge map is divided into small non-overlapping blocks. Each query block is correlated with a small neighborhood of blocks in the image edge map. This neighborhood is centered on the block that is in the same position as the query block. The correlation between a query and image block is defined as a sum of weights where there is one weight for each of six possible cases – query edge (blank) lined up with an image edge; query edge (blank) lined up with an image blank; and query edge (blank) lined up with a position that is off the side of the image. Computing this correlation is simply a matter of performing a pixel-by-pixel comparison of the query and image blocks. The maximum correlation over all image blocks in the local neighborhood is taken to be the correlation score for the query block. In other words the algorithm tries various shifts of the query block and chooses the shift that provides the best match. This allows the retrieval of images that have edges similar to those of the query but in slightly different positions. The correlation scores for the query blocks are summed and divided by the maximum possible sum for that particular query to get the edge similarity score.

We weighted the edge and color similarity scores equally; weighted each subarea equally; and used query edge/image edge, query edge/image blank, query edge/off image, query blank/image edge, query blank/image blank and query blank/off image weights of 10, -3, -1, -3, 1 and -1 respectively. These weights take into account the fact that a match between two edges is more important than a match between two blanks. In addition the sketch comparison algorithm uses 8 x 8 blocks and defines the local neighborhood of image blocks to be all blocks that are 4 or fewer pixels away from the center block. These values are used in [HK92]. Modified values did not produce significant improvements in retrieval performance.

2.4 Graphical user interface

Users can either use an existing image as a query or interactively *draw* their own query using a simple GUI. This GUI allows the user to draw edges and swatches of color. Once the user has selected an existing image or drawn her own image, the system constructs an edge map and a set of histograms. The edge map and histograms become the input to the query module.

3 Preliminary evaluation

A preliminary evaluation suggests several problems with the retrieval techniques. The system was evaluated with a database of forty-eight outdoor scenes that are sold as part of the Microsoft Scenes screen saver. These are the same scenes that were used to evaluate the edge detection algorithm. Figures 4–7 show the results of four specific queries that were made against the image database. Two queries involve only color and two queries involve only edges; queries that involve both color and edges seemed premature in light of the retrieval problems that were identified. In general the results of these four queries are typical of the retrieval behavior exhibited by the system.

Figure 4 shows the result of submitting an image to the retrieval system and requesting images that have a similar color distribution. A large version of the query image appears in figure 2. In this case the system compared each pair of subarea histograms using *histogram intersection* and averaged the subarea similarity scores to get the overall similarity score. The retrieval results are reasonably good in that three of the top five images have a composition similar to that of the query image – green foliage at the bottom, blue sky at the top and in two cases a gray mountainous region in the center. In addition no relevant images were ranked below the top ten. However two of the five images clearly do not belong. The image with rank 0.32 has no blue sky or gray mountain – it scores highly because it has lots of green at the bottom and several regions of small gray rocks at the top. The image with rank 0.44 has no blue sky or gray mountain – it scores highly because it has lots of green at the bottom.

This query illustrates the first two problems with our color retrieval mechanism. First there is no way to specify that the *absence* of a certain color from a region means that the image is irrelevant. For example we would like to specify that the absence of blue sky

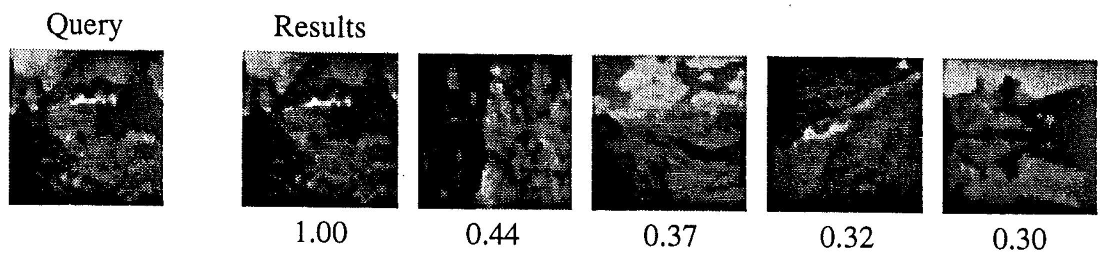


Figure 4: Here we are using the image shown in figure 2 as a *color query* – i.e. we want to find all images that have a similar color distribution. The five highest-ranked images are shown. As expected the highest-ranked image is the query image itself.

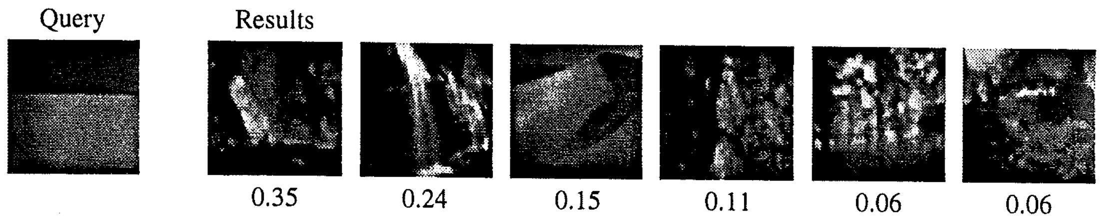


Figure 5: Here we are using a hand-drawn image as a *color query*. The query consists of a monocolor blue region at the top and a monocolor green region at the bottom – it is hoped that this query will retrieve images that have blue sky at the top and green foliage at the bottom. The six highest-ranked images are shown (all other images had a rank of zero).

means that the image is irrelevant. This can not be accomplished by increasing the weight of the two subareas at the top of the image since then we might retrieve images that have blue sky but no green grass. Rather we need to indicate that a negative result in one subarea overrides even the most positive result in another subarea. Second the algorithm does not provide effective localization. It does not take into account that a color should appear in a specific location within a subarea. For example, the gray mountain in the query image matches well against several widely separated regions of gray rock in the image with rank 0.32.

Figure 5 shows the result of submitting a hand-drawn query to the system. The user has drawn a monocolor blue region at the top and a monocolor green region at the bottom in an effort to retrieve images that have blue sky at the top and green foliage at the bottom. The system compared subarea histograms as before. The results are poor. Only six images have a nonzero similarity score and only the *lowest-ranked* of the six can be considered relevant to the query.

The top five images illustrate the same problem as before. There is no way to specify that the absence of a certain color from a region means complete irrelevance. The top five images are retrieved even though they either contain no blue at the top *or* contain no green at the bottom. However a more critical problem is that only one relevant image received a nonzero similarity score. The other relevant images – even the most relevant image which has an unbroken green field at the bottom, blue sky at the top and a flat horizon – had similarity scores of zero. The problem is that the histogram intersection algorithm performs *exact* color match. Only a single blue and a single green are used in the query; all images that do not contain the exact same shades of blue and green have an empty intersection with the query and therefore a similarity score of zero. This problem did not arise with the previous query since the previous query was an image taken directly from the database and contained enough different shades of blue, green and gray to ensure a good match with all relevant images. However the problem can arise with any real-world image that has large regions of uniform color.

Figures 6 and 7 are best considered together. Figure 6 shows the result of submitting an image to the system and requesting images that

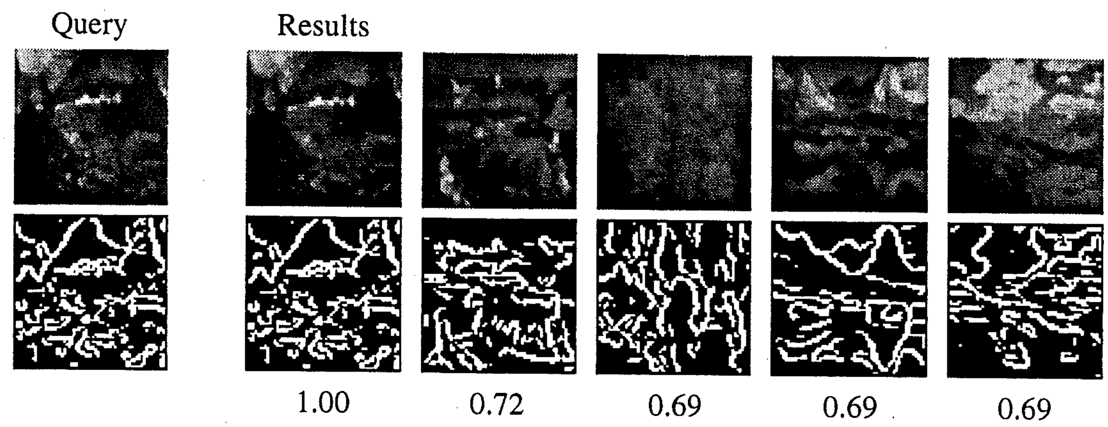


Figure 6: Here we are using the image shown in figure 2 as an *edge query* – i.e. we want to find all images that have a similar pattern of edges. The five highest-ranked images are shown. As expected the highest-ranked image is the query image itself.

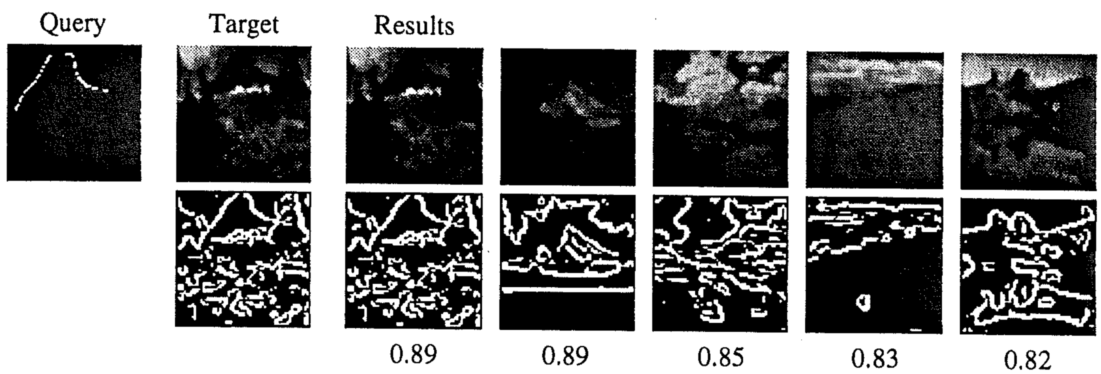


Figure 7: Here we are using a hand-drawn query as an *edge query*. It is hoped that this query will retrieve images that have mountain peaks in the center (particularly the "target" image that the author was looking at when he drew the query). The five highest-ranked images are shown.

have similar edge characteristics. The results are poor in the sense that there are relatively high ranks on irrelevant images and that there is little rank discrimination between relevant and irrelevant images. Figure 7 shows the result of submitting a hand-drawn query in an effort to retrieve images that contain mountains (particularly the indicated target image). In this case the weights in the sketch comparison algorithm were adjusted such that blank regions of the query were treated as "don't care" regions. Otherwise the query retrieves images that contain mainly blank space. The target image was successfully retrieved but again there are relatively high ranks on irrelevant images and there is little rank discrimination between relevant and irrelevant images. In addition the relevant images that score highly – aside from the target image – score highly by chance. The mountain edge in the query is lining up with foliage and cloud edges in the images.

There are three problems. The test collection is small so the results are partially an artifact of the fact that there are only a few images relevant to each query. However this problem is minor in comparison to the other two. The second problem is that many edge maps contain extraneous edges that are perceptually prominent on the basis of color but spurious when one considers domain knowledge. In addition some edge maps miss important edges that are prominent on the basis of texture or domain knowledge but not on the basis of color. For example the edge map for an image of trees through fog essentially contains

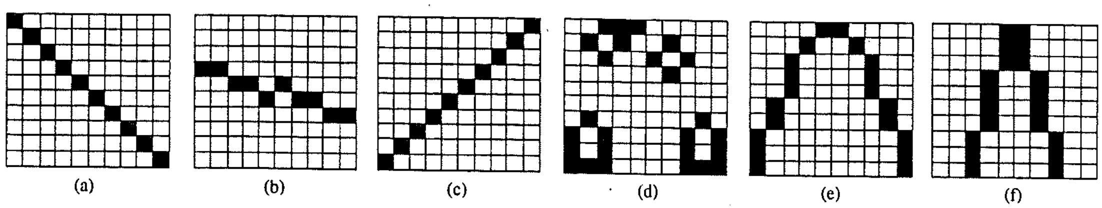


Figure 8: The fundamental weakness of the edge-based retrieval technique is that it performs a pixel by pixel comparison of the edge maps – therefore it reports that b is equally similar to a and c (in each case two edge pixels line up under the best possible shift) and that e is more similar to d than to f (thirteen edge pixels line up between d and e under the best possible shift but only eight edge pixels line up between e and f).

edges that trace each individual leaf cluster and no other edges. It would be more reasonable to have an edge that followed each tree trunk. Unfortunately the tree trunks blend into the fog and are not prominent on the basis of color alone. [HK92] and [KKOH92] have cleaner edge maps due to the nature of their images. They use a collection of paintings that tend to have far sharper color boundaries than our outdoor photographs.

The third and most critical problem is that the sketch comparison algorithm compares edge maps on a *pixel-by-pixel* basis. This leads to nonintuitive results as shown in figure 8. The edge maps shown in the figure were constructed by hand but are representative of some of the actual edge maps in the test collection – i.e. hillsides (a,b,c), mountains (e,f) and a rosebush (d). The problem is that the sketch comparison algorithm ignores high level features such as edge orientation, shape and connectivity. Instead it simply shifts query blocks around on top of the image and counts the number of pixels that line up. The result is that often one edge scores highly against several disconnected edges *or* an edge scores poorly against a highly similar edge since they are just different enough that only a few pixels line up. The method appears to perform well enough for retrieval of a *known* or *remembered* target image on the basis of a *well-drawn* sketch or for retrieval of similar images in a larger database than ours. These are the situations that were considered in [HK92] and [KKOH92] where sketch comparison was observed to provide reasonable performance. However it clearly does not provide reasonable performance for arbitrary queries against arbitrary databases.

4 Future work

4.1 Color

4.1.1 Weighting

The color retrieval technique has several weaknesses that were discussed in the evaluation section. First the color similarity score is just a weighted average of the subarea similarity scores so the user has only the coarsest level of control over query behavior. There is no way to specify that the absence of blue at the top means that the image is irrelevant, no way to specify that a subarea can contain blue *or* green and so on. This problem can be partially solved without abandoning the subarea scheme. The overall similarity score should be a nonlinear function of the subarea similarity scores so that the score for a subarea can have any desired effect on the overall score. In addition each subarea similarity score should be a nonlinear function of one *or more* histogram intersections.

4.1.2 Localization

Unfortunately there remains an inherent weakness in the subarea scheme. A subarea with blue on the left will score highly when matched against a subarea with blue on the right. Even worse a subarea with a large gray area will score highly when matched against a subarea with several small, disconnected gray areas. The problem is that the retrieval mechanism provides no color localization below the subarea level since it is just comparing histograms. The obvious solution is to significantly increase the number of subareas. A better solution is to leave the number of subareas unchanged and have the system perform

a direct comparison of the image and query if there is sufficient similarity between the histograms. Either approach allows the user to specify that a color should appear as a connected region with a certain shape and position. However we should not limit ourselves to saying that location is always important. The user should be able to specify that a color can appear anywhere (or anywhere within a certain region) without losing the ability to specify that the color should form a connected region with a certain shape. Neither of the approachs can efficiently support this behavior. The only recourse is to run multiple queries where each query is a transformation of the given query. This is computationally intractable for large databases or complicated queries.

To handle the problem of color localization – and the problem of specifying on a color by color basis whether location matters – some researchers use segments rather than subareas [CLP94, Na93b, GZCS94]. Each image is divided into segments such that each segment contains a single object or approximately a single uniform color. The color distribution of each segment is represented as a weighted centroid or as a histogram. The query is specified as a set of segments and the query segments are matched against the image segments. Features such as segment size, location and shape can be used in addition to segment color. Dividing an image into object-based segments can not be done automatically for general images so it is more attractive to segment the image into regions of uniform color. In either case the system must be prepared to match a single query segment against multiple image segments since the user might specify the query segments at a coarser resolution than the image segments (or vice versa). Since the user specifies query behavior on a segment-by-segment basis – e.g. this segment can appear anywhere within this region of the image; this segment must be at a fixed location but can contain any one of several colors, etc. – the segment-based approach eliminates the problem of color localization and allows fine-grained control over search behavior. We plan to move to a segment-based scheme for these reasons.

4.1.3 Exact color match

The third problem with the color retrieval technique is that histogram intersection performs exact color match. Instead we want inexact color match so that different shades of the same color will be considered similar. This is relatively simple in the case of histogram intersection. Rather than intersect a query bucket with an image bucket, the algorithm would intersect the query bucket with a neighborhood of image buckets. Different weights would be used for different buckets in the neighborhood. Note that reducing the number of histogram buckets along each color axis would not have the same effect. It would lead to inexact match for colors mapped to the center of buckets but not for colors mapped to the edge. In addition the user would not be able to specify inexact match on one query and exact match on the next.

4.1.4 Efficiency

Efficiency of histogram intersection is not a large concern at this point although response time will be poor for large databases. In this case there is a much more efficient version of histogram intersection called *incremental histogram intersection* that intersects the buckets in order of decreasing pixel count and stops if it determines that the similarity between the histograms can not be more than some threshold [SB91]. However, efficiency of the segment comparison algorithm – of which histogram intersection is just a part – will be critical when we move to segment-based retrieval. Reasonable efficiency will require an excellent representation for segment features as well as a hierarchical or cluster-based retrieval scheme. It should not be hard to implement a hierarchical scheme. For example, comparing the overall query histogram with the overall image histograms can immediately eliminate most of the database from consideration. Cluster-based retrieval will be more difficult since we must choose the color features used in the clustering process carefully. For example, if we cluster the images according to dominant color and then request images that contain a small region of blue in one corner, we have gained nothing and must search the entire database.

4.1.5 Other comparison metrics

Histogram intersection is one of a number of techniques for comparing color distributions. [CLP94] uses the color pair technique and has achieved good results in a small database. QBIC uses a matrix-based technique that takes the product of a difference histogram and a set of perceptual color distances [Na93b]. [GZCS94] reduces an entire histogram to a single integer key by transforming the histogram into a hyper-polygon and then taking a weighted sum of the angles and edge lengths. Unfortunately [Na93b] and [GZCS94] do not provide an analysis of retrieval performance. The technique of [GZCS94] will be exceptionally useful if it provides effective retrieval performance since then the first few levels of a hierarchical or cluster-based scheme could be reduced to integer comparison rather than histogram comparison. We plan to evaluate as many of these techniques as possible.

4.2 Edges

4.2.1 Edge detection

The edge detection algorithm does not construct good edge maps for every image. Some edge maps contain "extraneous" edges that provide no useful information for retrieval purposes even though they are perceptually prominent in the image. Other edge maps miss valid edges or portions of valid edges. Some of the extraneous edges arise due to "color noise" (i.e. a small group of pixels whose color is sharply different than that of their neighbors). More aggressive median filtering and a thinning procedure to strip out the shortest edges [KKOH92, HK92] should help with these extraneous edges. To address the rest of the extraneous edges and the missing edges, we need to explore different edge detection algorithms along with schemes for refining edge maps to get cleaner, more complete edge maps. Such schemes include tracing algorithms, reinforcement algorithms and heuristics. For example, one reasonable heuristic is that edges that cross a large portion of the image are more important than edges that wind around and around in a small section of the image. A complete discussion of edge detection is beyond the scope of this paper. A good – but slightly dated – overview appears in [Hor86]. In any event we need an edge detection algorithm that considers texture as well as color since some important edges are perceptually prominent on the basis of adjacent textures rather than adjacent colors. An alternative approach is to use two algorithms – one that uses color to detect edges and one that uses texture – and then either combine the two edges maps or use both edge maps in the retrieval process. Perhaps the most interesting approach is to apply several edge detection algorithms to each image and then automatically compare the resulting edge maps and select the best edge map to represent the image. None of these approaches will construct good edge maps for every image since some edges are perceptually prominent to humans only because humans *know* what objects are shown in the image. However it should be possible to significantly improve the edge maps without resorting to domain-specific knowledge or techniques.

4.2.2 Sketch comparison

The *sketch comparison* algorithm is the weak point in the retrieval system. Sketch comparison performs a pixel-by-pixel comparison to calculate the similarity between two edge maps. This leads to nonintuitive results as shown in figure 8. There are two contradictory problems. First, although blocks of pixels are allowed to shift in search of the best match, the pixels within the block are fixed so it is often impossible to line up more than a few pixels of highly similar edges. Second, it is possible for a single edge in one image to match portions of several different edges in a second image. The contradiction lies in the fact that increasing the mobility of blocks and pixels eases the first problem but makes the second far worse; restricting their mobility has the opposite effect.

It appears that pixel-by-pixel comparison must be abandoned in order to support arbitrary, *poorly drawn* queries against heterogeneous databases. An intermediate step is to keep the pixel-oriented edge maps but to make the query map edge-oriented – i.e. the query edges are represented as a set of paths rather than as an array of pixels. It is easy to obtain this set of paths when the user draws the query by hand; it is more difficult when the query is a sample image since edge trac-

ing or similar techniques must be used. The similarity score for a given *edge* in the query would be a function of the number of edge pixels that it covers in the image, whether those pixels are connected or disconnected, and how much the query edge needs to deform in order to move into position over those pixels. Using a deformation metric has been used successfully elsewhere – most notably the shape portion of the Photobook project [PPS94] and the "active snakes" that are often used in interactive outlining applications [Na93b]. A more drastic step is to make the image maps edge-oriented as well. Then the problem of comparing a query to an image would become a problem of comparing the feature values associated with the edges. Possible edge features are location, orientation, length, turning rate and so on. This would again require a more complicated edge detection algorithm but we might be using such an algorithm anyways for the purpose of cleaning extraneous edges out of the edge maps. In addition an entirely edge-oriented approach will make it easier to provide fine-grained query control – e.g. this edge can appear anywhere within this region of the image, this edge must be in this position but can have one of two orientations, etc. The segment-based approach to color retrieval has the same advantage. We plan to move to the deformation approach initially and then to the feature-based approach if necessary.

4.2.3 Efficiency

The remaining problem with *sketch comparison* is efficiency. The algorithm is faster than one might expect. However the algorithm performs a quarter of a million pixel comparisons just to compute the similarity between two 64 x 64 edge abstracts. The efficiency should improve with a careful implementation of the deformation or feature-based schemes mentioned above. However it seems clear that a hierarchical or cluster-based retrieval mechanism will be needed for larger databases. We are attempting to determine which features of the edges will be useful in forming hierarchies or clusters. At a minimum images can be grouped according to which regions contain no edges. However much more sophisticated schemes are possible.

4.3 System evaluation

The preliminary evaluation used a test collection of only forty-eight images. We must move to a larger test collection in order to perform a detailed evaluation of the system and of the extensions that have been proposed. A larger test collection is particularly essential for examining the time requirements of the retrieval mechanisms. We are currently building a large image corpus that can be used for this and other image retrieval work. However the small test collection should not be abandoned since it is useful for exploring the situation in which there are only a few relevant images per query and in which images tend to be highly different from each other.

In addition we need to evaluate the graphical user interface in terms of the ease with which the user can construct an appropriate query. This issue will become more acute as we incorporate additional features such as texture and shape into the system since the user must be able to easily combine multiple features into a single query and must be able to easily specify the characteristics of and the relationships between the various parts of the query. The current GUI is simplistic and will need substantial reengineering to achieve these goals.

4.4 Texture, shape and other features

Color and edges are two of a wide range of image features that have been used in content-based retrieval. One common feature is texture. [SC94] uses quad-tree segmentation to divide an image into blocks of approximately uniform texture. Feature vectors for the textures are computed from mean and variance measures produced by a QMF wavelet decomposition. The user queries the database by selecting the desired texture from a set of prototypical textures. The approach works well on a collection of synthetic images but has not been tested on real-world images [SC94].

The QBIC project [Na93b, Na93a] allows texture-based retrieval as well. It uses three features. *Coarseness* measures the scale of the texture and is computed with moving windows of several sizes; *contrast* describes the "vividness" of the texture and is computed from a gray-level histogram; *directionality* measures

whether the texture has a "favored" direction and is computed from the gradient directions [Na93b]. The authors note that other texture features were either too expensive to compute or were ill-suited to heterogeneous collections of images [Na93b]. The user queries the database by providing a sample of the desired texture. Unfortunately the authors do not provide an analysis of retrieval performance.

A second common feature is the shape of the objects in the image. QBIC uses a combination of area, circularity, eccentricity, major axis orientation and moment invariants to represent a shape [Na93b]. [GZCS94] uses just circularity and major axis orientation. In QBIC the user draws the desired shape. Then the system computes the features of the query shape and matches the features against the features of each image shape. In [GZCS94] the user does not draw the shape but rather specifies the values of the two shape parameters directly. Unfortunately both systems require that the images be segmented along *object* boundaries. QBIC resorts to a manual approach in which a human manually outlines the desired shapes using an interactive "shrink-wrap" utility [Na93b]. [GZCS94] segments the images automatically on the basis of color. However postprocessing is required to recover from over-segmentation. The postprocessing is not described but is probably manual. Both authors provide no analysis of retrieval performance.

Texture and shape will be incorporated into our system as soon as the weaknesses of the color and edge retrieval mechanisms have been addressed. Shape will be more difficult to incorporate since most approaches rely on segmenting images along object boundaries. However – as in the segment-based approach to color retrieval – it should be possible to develop a representation that allows a single shape in the query to match multiple shapes in the image (or vice versa). Then the images can be segmented on the basis of color and texture rather than along object boundaries.

The use of image features such as texture and shape will improve retrieval performance. However text should not be ignored since many images must have text associated with them anyways – e.g. captions for photographs in a book – and many queries are impossible to answer without examining the text. For example, finding all photographs taken in Paris is impossible without looking at the photograph captions. Allowing the user to search all available text will be a critical addition to the retrieval system despite the problems with text-only retrieval. A simple text retrieval mechanism will be incorporated soon. In addition some databases have other kinds of manual or automatic annotations. The system should be able to search these annotations as well. This means that the system must be extensible. A package of routines would be written to handle a particular kind of annotation; the retrieval system would be notified of their existence and would use the routines to search that particular kind of annotation. The package of routines would have to conform to some standard interface.

5 Conclusion

This paper has presented a content-based image retrieval system that does not use domain-specific knowledge or techniques. Instead the system retrieves images on the basis of their color distributions and edge characteristics. Two existing retrieval techniques – *histogram intersection* and *sketch comparison* – are used to compare the color distributions and edge maps. The performance of the system shows some promise but falls far short of the performance that is required for practical electronic publishing. Histogram intersection and sketch comparison are not general enough. They perform poorly when removed from the context in which they were developed. Thus work to improve content-based retrieval systems such as ours must proceed on several levels. The low-level comparison techniques need to perform well against a wide variety of images; several approaches to improving histogram intersection and sketch comparison have been explored here and elsewhere in the literature. However all low-level techniques have weaknesses that can not be eliminated. The higher-level retrieval mechanisms and the the graphical user interface must be flexible, powerful and easy to use so that these weaknesses affect the user as little as possible. In short the low-level techniques can provide a large set of potentially relevant images as long as the GUI allows the user to quickly sift through the set and easily reformulate the query if desired.

The GUI is perhaps the most critical part of a successful retrieval system. Addressing both the low-level techniques and the GUI should lead to a system that can provide effective retrieval performance for most image databases *even in the absence of domain knowledge*; in addition such a system should significantly reduce the amount of domain knowledge that is required for effective retrieval performance against other databases.

6 Acknowledgements

Many thanks to Jing Feng and Professor Fillia Makedon for useful discussions; to my advisor, Professor George Cybenko, for his encouragement and support; and, as always, to Jennifer and Stephen Gray for reminding me that there is life outside graduate school.

References

[Ben94] Jim Benson. Searching for stock photos online. *MACWORLD*, pages 124–127, August 1994.

[CLP94] Tat-Seng Chua, Swee-Kiew Lim, and Hung-Keng Pung. Content-based retrieval of images. In *Multimedia 94*, San Francisco, Calfiornia, 1994.

[FVDFH91] James D. Foley, Andries Van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading, Massachusetts, second edition, 1991.

[Gra95] Robert S. Gray. Content-based image retrieval: color and edges. Research Report PCS–TR95–252, Department of Computer Science, Dartmouth College, Hanover, New Hampshire, 1995.

[GZCS94] Yihong Gong, Hongjiang Zhang, H. C. Chuan, and M. Sakauchi. An image database system with content capturing and fast image indexing abilities. In *Proceedings of the International Conference on Multimedia Computing and Systems*, Boston, Massachusetts, 1994.

[HK92] Kyoji Hirata and Toshikazu Kato. Query by visual example. In *Advances in Database Technology EDBT 1992, Third International Conference on Extending Database Technology*, Vienna, Austria, 1992.

[Hor86] Bertjold Klaus Paul Horn. *Robot Vision*. The MIT Press, Cambridge, Massachusetts, 1986.

[KKOH92] Toshikazu Kato, Takio Kurita, Nobuyaki Otsu, and Kyoji Hirata. A sketch retrieval method for full color image databases. In *International Conference on Pattern Recognition (ICPR)*, The Hague, The Netherlands, 1992.

[Na93a] Wayne Niblack and all. The QBIC project: Querying images by content using color, texture and shape. *SPIE*, 1908:173–187, 1993.

[Na93b] Wayne Niblack and all. The QBIC project: Querying images by content using color, texture and shape. Research Report RJ 9203 (81511), IBM Research Divison, Almaden Research Center, San Jose, California, 1993.

[PPS94] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: Tools for content-based manipulation of image databases. In *SPIE Storage and Retrieval Image and Video Databases II*, San Jose, Calfiornia, 1994. Also available as: M.I.T. Media Laboratory Perceptual Computing Technical Report No. 255.

[SB91] Michael J. Swain and Dana H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11–32, 1991.

[SC94] John R. Smith and Shih-Fu Chang. Quad-tree segmentation

for texture-based image query. In *Multimedia 94*, San Francisco, Calfiornia, 1994.

[SF91] Thomas M. Strat and Martin A. Fischler. Context-based vision: Recognizing objects using information from both 2-D and 3-D imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1050–1065, October 1991.

[Swa93] Michael J. Swain. Interactive indexing into image databases. *SPIE*, 1908:95–103, 1993.

Structural Queries in Electronic Corpora

Daniela Rus
Department of Computer Science
Dartmouth College
Hanover, NH 03755
rus@cs.dartmouth.edu

James Allan
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
allan@cs.umass.edu

Abstract

We present a methodology for automatically constructing structural hyperlinks in electronic technical corpora. A structural hyperlink connects components of a document that have specified structural properties with word-based content similarity. Our approach enables queries that may be posed in terms of keywords, as well as structural segments such as definitions, figures, *etc.*

1 Introduction

In today's environment of distributed electronic libraries, documents have an inherently multi-media structure, as they may include any combination of text, graphics, video, and audio. We are interested in the *structure-based* searching and indexing of electronic documents, in order to provide support for content-based retrieval. Our notion of structure is general and refers to any regularity encountered in the data. For example, in the domain of technical documents, we would like to be able to retrieve **documents** with **pictures** of mobile robots, or **documents** with **graphs** describing the performance of file systems.

In our previous work [RS95b, RS95], we have presented robust algorithms for recognizing the underlying structure of documents in terms of their logical visual components (tables, graphs, sections, *etc.*) These structures encode information about content and can provide the basis for the on-line creation of browsing tools and semantic links between electronic documents. For example, Figure 1 shows a vastly reduced picture of the pages in a paper. Even though none of the words is legible, a great deal can be said about this paper based purely upon the layout: *e.g.*, the first page contains title and author information, section breaks are identifiable, figures and tables are scattered throughout. Documents with figures and graphs have a surprising amount of information encoded in their layout on the page. Page layout formats have evolved to broadly utilized standards and we propose to explore the benefits of layout organization standards for information retrieval.

Electronic documents have a multiplicity of content and layout structures that are not exploited by traditional keyword-oriented retrieval methods. Here, we focus on structures that capture the human conventions for typesetting papers. The structural types we consider are: titles, authors, institution, sections, definitions, figures, figure captions, theorems, proofs, paragraphs, itemized lists, and tables.

Like keywords, structures can also function as indexes for cataloging electronic documents. In earlier work we have discussed our methodology for structure-based information gathering.[RS95] We have also presented algorithms for automatically segmenting electronic documents in structural components as well as algorithms for classifying the types of the structural components.[RS95b] In this paper we present an approach to combining structural indexes with keyword indexes to enrich the class of retrieval queries.

1.1 Previous Work

Our work draws on previous work in two distinct areas: information retrieval and automated document structuring.

Current information retrieval systems are primarily word- or word-group-driven.[SM83, Sal89, Tur90] The vector space model used in the Smart system[Sal91] has been used primarily for document retrieval, but is equally effective for document comparison,[SA93] and can also be used for the automatic identification and description of hypertext links.[All95]

The goals of the document structuring community are to identify the key constituents of a document image (sections, paragraphs, pictures, *etc.*) from its layout and to represent the logical relationship between

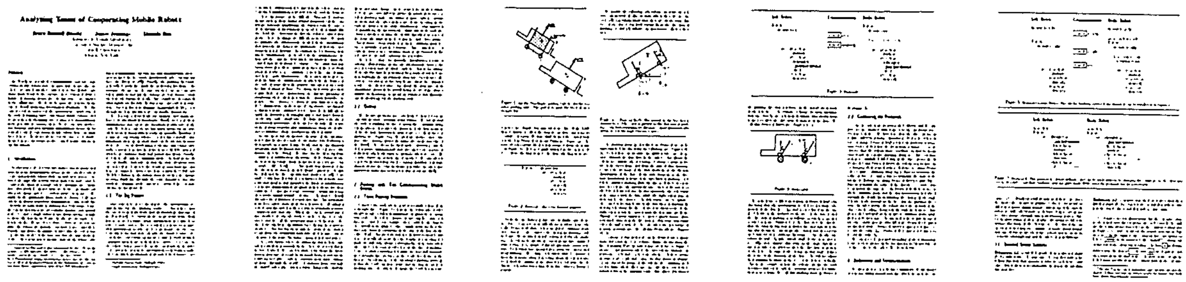


Figure 1: A zoomed-out view of an article on coordinated pushing with mobile robots. The first page has the title/author information represented as centered text. Page three has an itemized list. Pages three and four contain polygonal drawings of boxes. Pages four and five contain robot protocols presented in tabular form.

these constituents.

Document structuring is usually done in two phases. In the first phase, the location of the blocks on the page is determined. In the second phase, the blocks are classified and the logical layout of the document is calculated. Previous work on block segmentation of documents include [JB92, NSV92, WS89]. Previous work on classifying and logically relating blocks includes [TA92, TSKK88, FNK92, NSV92, MT*91, WS89].

It is also possible to use vector space comparison of document passages to determine topic and subtopic structures of a document, based upon or independent of its layout structure.[HP93, SABS94, SS94]

In our own work, we have introduced robust algorithms with performance guarantees for segmentation as well as classification.[RS95b] Our vision for information access with structure-based information agents has been discussed in [RS95].

Structure has been identified as being a source of knowledge in other recent work.[FMSW93] In that study, they described a system that relies on SGML marked-up documents to support structured queries. The mark-up identifies the sections and paragraphs that comprise a document. These finer structural units can be referred to in a query. Other related work includes [KM93] that describe a tree inclusion grammar for retrieving information from collections of hierarchical text.

2 Constructing Structural Hyperlinks

Imagine reading an ICRA[1] paper that contains a cryptic definition of an "immersed sensor system". Starting from this definition, it is desirable to allow a user to find text explaining the definition in greater detail or perhaps even figures which illustrate it, whether the text or figures are within the current paper or in a larger collection of related papers. The process of building such links from the definition can be automated by using a combination of algorithms for segmenting, classifying, indexing, and hyperlinking.

Our data consists of a collection of technical papers available in a variety of formats: PostScript, LaTeX source, ASCII, *etc.* In order to execute tasks such as finding an explanation of a definition in an electronic collection, each document requires the following preprocessing: (1) segmentation, to highlight the structural components; (2) classification, to associate a type with each structural segment identified in (1); and (3) indexing, to catalog document components of similar type. Note that steps (1) and (2) can be used automatically to mark-up the original document in an SGML-like fashion.

We have built a system for information retrieval that uses text as well as type information for each block of text to support structural queries. Our system has three major components: the *segmenter*, which is a program that automatically divides the data into units; the *classifier*, which is a program that automatically labels the units produced by the segmenter with document structural types; and the *re-*

[1] International Conference on Robotics and Automation.

triever, which is a program that finds the responses to a user query. The collection used for retrieval consists of the source documents which have been divided by the segmenter and labelled by the classifier. The Smart system is extended to take block types into account.

We illustrate the segmentation and classification using a PostScript version of the paper with the cryptic definition.[DJR94] A selected portion of the paper is depicted on the left in Figure 2, including a figure at the top, a text paragraph continued from the previous page, a section, and finally a subsection including 2 definitions.

A series of filters are applied to the PostScript file as follows. First, `ps2ascii` is used to convert it to ASCII, preserving as much of the layout as possible. Next, a block segmentation and classification filter is applied to identify blocks of text and to annotate them as much as possible.[2] The result of applying this sequence of filters to the original PostScript is shown on the right of Figure 2. Each block of text has been surrounded by SGML-style annotations describing the block as much as possible.

The segmentation and classification filter is designed to work primarily on technical papers and uses knowledge of that domain in order to identify correctly the following structural components of electronic documents: *sections, subsections, theorems, proofs, definitions, figures, figure captions, paragraphs, itemized lists*, and *tables*. After these blocks have been identified, the annotation can be incorporated into an indexed version of the document, allowing paragraphs, lists, tables, and other layout structures to be referenced directly in a query.

This preprocessing also generates a hierarchy of representations for each document. That is, each representation is a partition of the document at a different level of granularity—*e.g.*, a technical paper can be represented as a collection of sections, as well as a collection of (typed) paragraphs (*e.g.*, a definition is a paragraph of type *definition*).

Corresponding to such a hierarchy of representations, we can construct a hierarchy of structural indices for the corpus. For example, we have an index of definitions, an index for sections, and index for figure captions, *etc.* This hierarchy of indices drives retrieval in the manner of the Smart system.

The structural tags assigned to text blocks can be used during retrieval in several ways: to limit the class of items retrieved, to help the user see the relationships between retrieved items, and to affect the manner in which retrieved items are displayed.

2.1 Limited displayed items

The first of these uses is easy to understand since it requires only a slight modification to a query language. For example, a user looking at a figure caption from the sample ICRA paper and wanting more details might select that caption and generate a query such as: like this:

```
#TYPE(PARAGRAPH) Sensor system P.I(QS),
a circuit for Protocol I (QS). This
circuit shows one possible
implementation of the protocol.  Figures
8-9 do not show how to handle loss of
contact (i.e., the (break?)  case), but
this circuitry is easily added, and is
the same for both P.I(QS) and P.II.
```

In this case, the operator "#TYPE" is used to select particular types of text blocks as useful for retrieval. The rest of the query is used to identify which of the "paragraph" blocks are to be retrieved.

2.2 Seeing block relationships

The second use of annotated blocks is illustrated in Figures 3. In this case, the query listed above was given with no type requirement specified. However, the system was given a command which displays the retrieved parts of the document along with their types. Only the document containing the figure caption query is presented, represented by the curved bar. Blocks are represented by shaded segments of the bar, their position and lengths proportional to the position and length of the block within the document.[3] Starting with the caption query, the system found several paragraphs of text describing the object in the figure, as well as a theorem referring to the figure, its proof, and an application. The user could then choose which one or more of those parts and follow the appropriate "hyperlink".

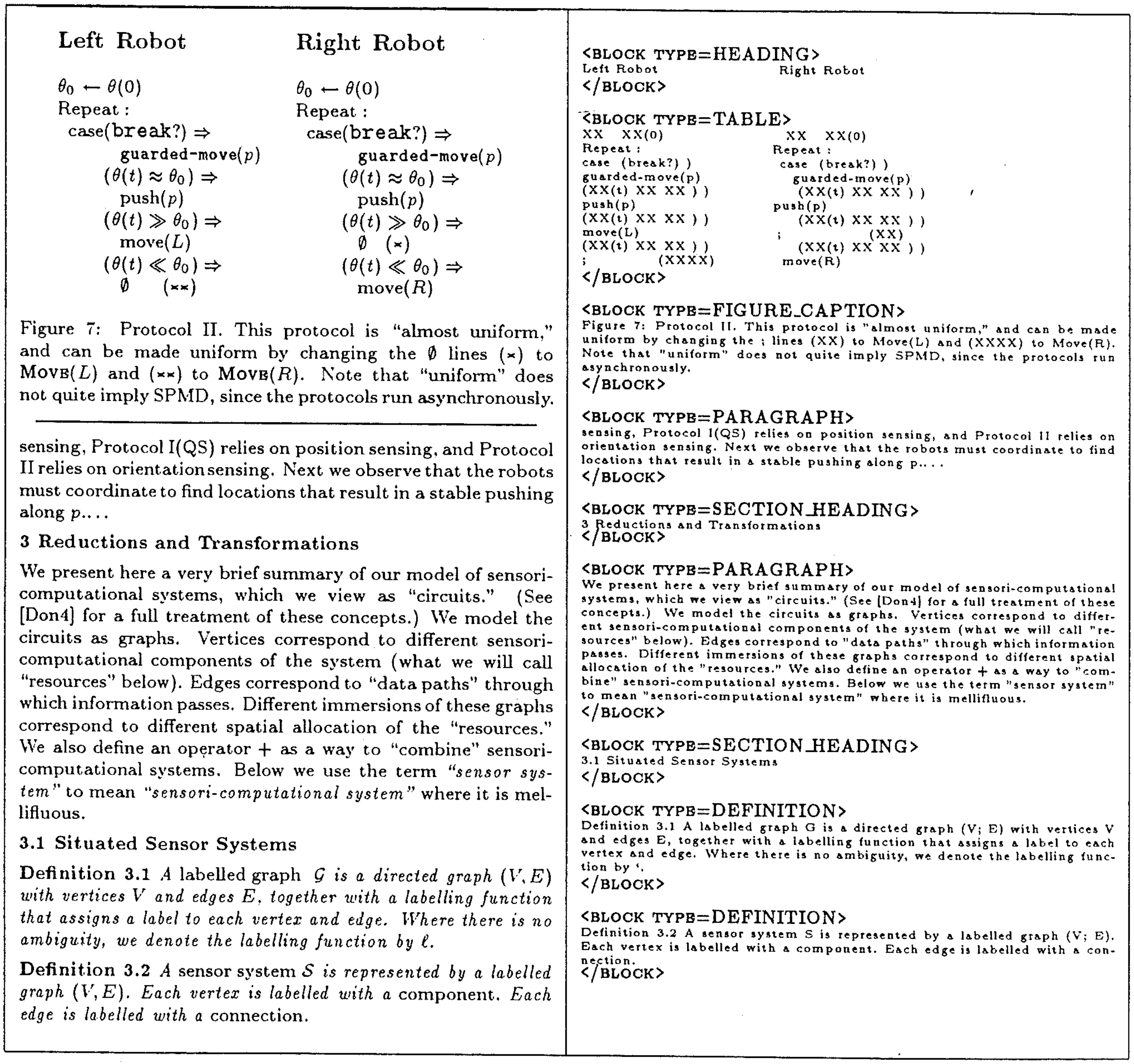
Figure 4 shows a similar map of blocks, types, and relationships to a query block, but the retrieval is extended to include multiple documents.

2.3 Types affect display

The third use of tagged blocks is also illustrated in both Figures 3 and 4. If the user should elect to dis-

[2] This filter is based upon the table extraction filter described in [RS95], but has been extended to use pattern matching techniques to identify other types of blocks.

[3] Additional annotations were added manually to the system-generated figure for presentation purposes.



Left Robot

$\theta_0 \leftarrow \theta(0)$
Repeat :
case(break?) $\Rightarrow$
guarded-move(p)
$(\theta(t) \approx \theta_0) \Rightarrow$
push(p)
$(\theta(t) \gg \theta_0) \Rightarrow$
move(L)
$(\theta(t) \ll \theta_0) \Rightarrow$
$\emptyset$ (**)

Right Robot

$\theta_0 \leftarrow \theta(0)$
Repeat :
case(break?) $\Rightarrow$
guarded-move(p)
$(\theta(t) \approx \theta_0) \Rightarrow$
push(p)
$(\theta(t) \gg \theta_0) \Rightarrow$
$\emptyset$ (*)
$(\theta(t) \ll \theta_0) \Rightarrow$
move(R)

Figure 7: Protocol II. This protocol is "almost uniform," and can be made uniform by changing the $\emptyset$ lines (*) to MOVB(L) and (**) to MOVB(R). Note that "uniform" does not quite imply SPMD, since the protocols run asynchronously.

sensing, Protocol I(QS) relies on position sensing, and Protocol II relies on orientation sensing. Next we observe that the robots must coordinate to find locations that result in a stable pushing along p....

3 Reductions and Transformations

We present here a very brief summary of our model of sensori-computational systems, which we view as "circuits." (See [Don4] for a full treatment of these concepts.) We model the circuits as graphs. Vertices correspond to different sensori-computational components of the system (what we will call "resources" below). Edges correspond to "data paths" through which information passes. Different immersions of these graphs correspond to different spatial allocation of the "resources." We also define an operator + as a way to "combine" sensori-computational systems. Below we use the term *"sensor system"* to mean *"sensori-computational system"* where it is mellifluous.

3.1 Situated Sensor Systems

Definition 3.1 *A labelled graph $\mathcal{G}$ is a directed graph (V, E) with vertices V and edges E, together with a labelling function that assigns a label to each vertex and edge. Where there is no ambiguity, we denote the labelling function by ℓ.*

Definition 3.2 *A sensor system $\mathcal{S}$ is represented by a labelled graph (V, E). Each vertex is labelled with a component. Each edge is labelled with a connection.*

```
<BLOCK TYPE=HEADING>
Left Robot                      Right Robot
</BLOCK>

<BLOCK TYPE=TABLE>
XX   XX(0)                      XX   XX(0)
Repeat :                        Repeat :
case  (break?) )                 case  (break?) )
guarded-move(p)                    guarded-move(p)
(XX(t) XX XX ) )                    (XX(t) XX XX ) )
push(p)                         push(p)
(XX(t) XX XX ) )                    (XX(t) XX XX ) )
move(L)                         ;            (XX)
(XX(t) XX XX ) )                    (XX(t) XX XX ) )
;           (XXXX)               move(R)
</BLOCK>

<BLOCK TYPE=FIGURE_CAPTION>
Figure 7: Protocol II. This protocol is "almost uniform," and can be made
uniform by changing the ; lines (XX) to Move(L) and (XXXX) to Move(R).
Note that "uniform" does not quite imply SPMD, since the protocols run
asynchronously.
</BLOCK>

<BLOCK TYPE=PARAGRAPH>
sensing, Protocol I(QS) relies on position sensing, and Protocol II relies on
orientation sensing. Next we observe that the robots must coordinate to find
locations that result in a stable pushing along p....
</BLOCK>

<BLOCK TYPE=SECTION_HEADING>
3 Reductions and Transformations
</BLOCK>

<BLOCK TYPE=PARAGRAPH>
We present here a very brief summary of our model of sensori-computational
systems, which we view as "circuits." (See [Don4] for a full treatment of these
concepts.) We model the circuits as graphs. Vertices correspond to differ-
ent sensori-computational components of the system (what we will call "re-
sources" below). Edges correspond to "data paths" through which information
passes. Different immersions of these graphs correspond to different spatial
allocation of the "resources." We also define an operator + as a way to "com-
bine" sensori-computational systems. Below we use the term "sensor system"
to mean "sensori-computational system" where it is mellifluous.
</BLOCK>

<BLOCK TYPE=SECTION_HEADING>
3.1 Situated Sensor Systems
</BLOCK>

<BLOCK TYPE=DEFINITION>
Definition 3.1 A labelled graph G is a directed graph (V; E) with vertices V
and edges E, together with a labelling function that assigns a label to each
vertex and edge. Where there is no ambiguity, we denote the labelling func-
tion by '.
</BLOCK>

<BLOCK TYPE=DEFINITION>
Definition 3.2 A sensor system S is represented by a labelled graph (V; E).
Each vertex is labelled with a component. Each edge is labelled with a con-
nection.
</BLOCK>
```

Figure 2: Portion of segmented and classified document

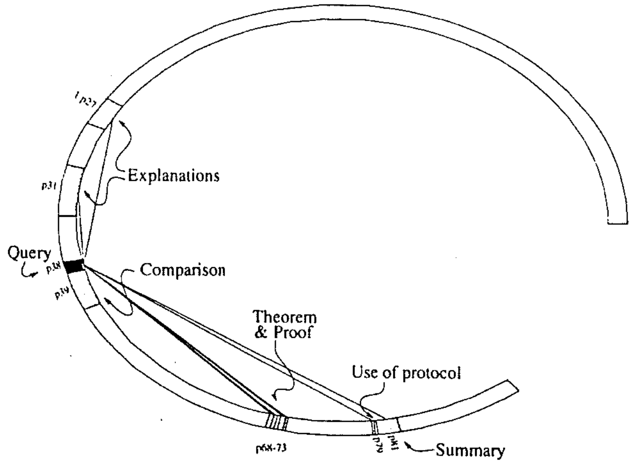


Figure 3: Structural hyperlinking for centralized information access. The circular arc represents a document. Various of its layout pieces are highlighted and edges are drawn between related parts.

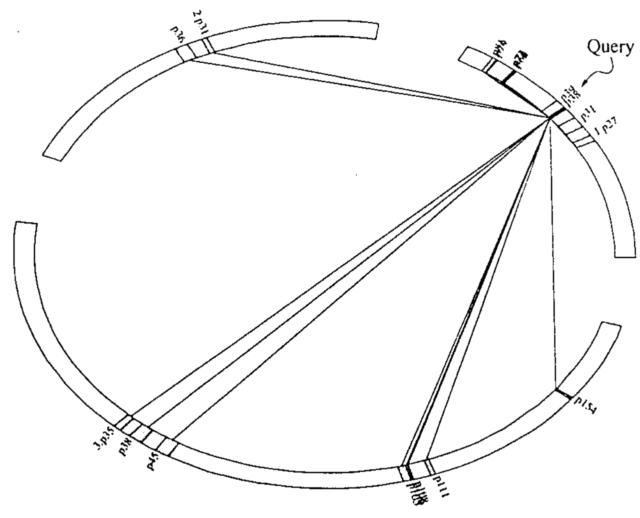


Figure 4: Structural hyperlinking for centralized information access. The circular arcs represents two documents. Various of their layout pieces are highlighted and edges are drawn between related parts.

play figure captions which are linked to the starting point, the system could launch a graphics display routine to display the figure along with the caption; if a "paragraph" is being selected, no such display routine is needed.

Several components of the graph layout provide a user with clues about what to expect: the size of blocks (represented by the width of the corresponding nodes in the graph), the relative location of the block in the document (the position of the node in the graph), the type of the block (which can be shown by means of color: each block type is presented in a different color), and connections (represented as edges in the graph). We believe that "a picture is worth a thousand words": that is, graphical summaries can be parsed faster and easier by humans than their textual counterparts. User studies are needed to fully validate this point in this context.

3 Discussion

Our research agenda is to develop and prototype a methodology for conceptual retrieval tasks in large, heterogeneous, distributed electronic libraries. We consider the key question to be how to associate information with content. We believe that we can get closer to our challenging long-term goal by considering hybrid similarity algorithms for the automatic generation of links across components of the corpus exhibiting structure. In this paper we present an application that validates this point of view. Our methodology has implications for the on-line browsing and searching of electronic corpora.

We have described an approach to gathering information that assists in understanding a figure caption (and can similarly make sense of a cryptic definition, or any other query that has some structure information in it) when it is possible to generate a centralized, multi-level index of the corpus. When the index does not exist or is distributed, the problem becomes more difficult as resource discovery becomes a sub-task.[GGT93] Our current project is to implement structure-based queries in the form of information agents operating across the World Wide Web environment.

References

[All95] J. Allan. *Automatic hypertext construction*, PhD thesis, Department of Computer Science, Cornell University, January 1995.

[DJR94] B. Donald, J. Jennings, and D. Rus. Analyzing Teams of Cooperating Mobile Robots. In *Proceedings of the International Conference on Robotics and Automation*, San Diego, 1994.

[FNK92] H. Fujisawa, Y. Nakano, and K. Kurino. Segmentation

methods for character recognition: from segmentation to document structure analysis. *Proceedings of the IEEE*, vol. 80, no. 7, 1992.

[FMSW93] M. Fuller, E. Mackie, R. Sacks-Davis, and R. Wilkinson. Structured answers for large structured document collections. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 205-213, 1993.

[GGT93] L. Gravano, H. Garcia-Molina, and A. Tomasic. The Efficacy of GlOSS for the Text Database Discovery Problem. Technical Report no. STAN-CS-TN-93-01, Computer Science Department, Stanford University, 1993.

[HP93] M. Hearst and C. Plaunt. Subtopic Structuring for Full-Length Document Access. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 59-68, 1993.

[JB92] A. Jain and S. Bhattacharjee. Address block location on envelopes using Gabor filters. *Pattern Recognition*, vol. 25, no. 12, 1992.

[KM93] P. Kilpelainen and H. Mannila. Retrieval from hierarchical texts by partial patterns. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 214-222, 1993.

[MT*91] M. Mizuno, Y. Tsuji, T. Tanaka, H. Tanaka, M. Iwashita, and T. Temma. Document recognition system with layout structure generator. *NEC Research and Development*, vol. 32, no. 3, 1991.

[NSV92] G. Nagy, S. Seth, and M. Vishwanathan. A prototype document image analysis system for technical journals. *Computer*, vol. 25, no. 7, 1992.

[RS95] D. Rus and D. Subramanian. Information Retrieval, Information Structure, and Information Agents. Submitted.

[RS95b] D. Rus and K. Summers. Using whitespace for automated document structuring. To appear in *Advances in digital libraries*, N. Adam, B. Bhargava, and Y. Yesha, editors. Springer-Verlag, Lecture Notes in Computer Science, 1995.

[Sal89] G. Salton. *Automatic Text Processing: the transformation, analysis, and retrieval of information by computer*, Addison-Wesley, 1989.

[Sal91] G. Salton. The Smart document retrieval project. In *Proceedings of the Fourteenth Annual International ACM/SIGIR Conference on Research and Development in Information Retrieval*, pages 356-358.

[SM83] G. Salton and M. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.

[SA93] G. Salton and J. Allan. Selective text utilization and text traversal. In *Hypertext '93 Proceedings*, pages 131-144, Seattle, Washington, 1993.

[SABS94] G. Salton, J. Allan, C. Buckley, and A. Singhal. Automatic analysis, theme generation, and summarization of machine-readable texts. *Science*, 264:1421-1426, June 1994.

[SS94] G. Salton and A. Singhal. Automatic text theme generation and the analysis of text structure. Technical Report TR94-1438, Cornell University, Department of Computer Science, July 1994.

[TSKK88] Y. Tanosaki, K. Suzuki, K. Kikuchi, and M. Kurihara. A logical structure analysis system for documents. *Proceedings of the second international symposium on interoperable information systems*, 1988.

[TA92] S. Tsujimoto and H. Asada. Major components of a complete text reading system. In *Proceedings of the IEEE*, vol. 80, no. 7, 1992.

[Tur90] H. Turtle. Inference networks for document retrieval. PhD thesis. University of Massachusetts, Amherst, 1990.

[WS89] D. Wang and S. Srihari. Classification of newspaper image blocks using texture analysis. *Computer Vision, Graphics, and Image Processing*, vol. 47, 1989.

Hypermedia Browsing and the Online-Publishing Process

Klaus Süllow, Rainer Pagé
GMD-IPSI, Darmstadt, Germany
{suellow,page}@darmstadt.gmd.de

Abstract

Editorial work in online publishing environments poses new technological requirements: how can an editorial group manage their cooperative work on a fast-changing information network and, even more important, how can they be enabled to manage the eventually published product containing interactive links to other publications.

The BWON browser system presented in this paper gives some support to handle these questions. Its client-server architecture helps to synchronize group work, and designed information filters control product quality requirements. BWON is an extension of the MultiMedia Forum editorial system and supports online publishing on the WorldWideWeb.

1 Introduction

In recent years systems for computer-based publishing have been developed and introduced into the marketplace [10]. The more sophisticated of these systems are aiming at product integration: one editorial process results in a diversity of end products (e. g. books, CD-ROMs, newspapers) with the same or similar content. At first glance publication via online access, which is just one more of these end products, apparently fits seamlessly into this product integration conception.

But if this new distribution path is combined with the idea of distributed hypermedia as in the case of the WorldWideWeb (W^3) [3], a new challenge to publishing work will arise: Online publishing under such circumstances means the dissemination of a hyperweb into an existing hyperweb requiring tools for managing the references between a publishing product and any other published information, which henceforth can be followed interactively.

For this aim the **BWON** system was built and integrated into the editorial environment of IPSI's online journal **MultiMedia Forum** (MMF) [13]. BWON provides support for browsing in the W^3 and from this point of view it can be regarded as an extension of Mosaic [11]. But additionally the tool allows access to the MMF editorial database and thus enables the editorial staff to manage the connections between the MMF database and the W^3.

In chapter 2 we present our conception of the online-publishing model and show the most important technical requirements which can be concluded from this model. After giving a short overview of the MultiMedia Forum (chapter 3) we describe the BWON system in detail in chapter 4.

2 A Model for Online Publishing

2.1 The Term "Online Publishing"

While today the term "online publishing" mostly means any provision of published information for network access, we qualify the term by including

- the availability of global addressing schemes like W^3's Universal Resource Locators (URL) [2] or HyTime's location addresses [9],
- the use of a hypermedia document format, e.g. HTML (Hypertext Markup Language) [4] and
- the possibility for the readers to enhance the originally published documents by adding their own contributions. Depending on the concrete publication product this may imply some complications to the editorial process (cf. section 2.2).

Since every publication appears in a *context of previously existing information* by citations, references, reviews etc., with online publishing the editor can additionally provide interactive access to this referred information, increasing the importance of such references for the publication. In other words, online publishing means to insert a hypermedia web into an existing hypermedia network.

Publishing products exceeding a certain size are always results of group work. Systems for cooperative hypermedia authoring [12] exist, but they are not designed for online publishing. Most important, they lack ensuring

a consistent view of the external online information either referenced by the product or at least playing a role in the editorial process.

To sum up, a system for online-publishing support has to provide a tool for management of distributed hypermedia structures by a group of editors. To give a more precise idea of this requirement we examine some details.

2.2 Online-Product Quality

Product quality can be measured by determining to what extent certain requirements are fulfilled. That is, the term "quality" can be reduced to requirements. In contrast to paper documents, electronic documents require a certain technical equipment. Thus **online-product quality** of a publication does not depend solely on its contents (including structure and layout), but also on the grade of fulfilling technical requirements, which in turn might influence the contents. We distinguish three groups of quality requirements:

1. **Technical quality requirements.** Perfect fulfillment of these requirements guarantees the reader best presentability of all parts of the product. Examples are the size (in case of small computers with narrow bandwidth connections) or the information type of a document (e.g., a requirement might be "no audio").

2. **Structural quality requirements.** To reduce the "lost in hyperspace" problem, some rules constricting the hyperweb structure should be imposed. E.g., all references to external documents should lead to dead ends after some navigation steps (otherwise the editor-in-chief has to approve the reference) or links to competing providers have to be avoided (not a trivial requirement, because the competitor might not be within reach until some navigation steps have been carried out). Structural quality is particularly endangered by the participation of the readers in contributing, because at best weak editorial guidelines can be imposed on the readers. Thus the editors need support (a) in detecting structural deficiencies in the readers' contributions and (b) in removing this lack of quality without falsifying the message.

3. **Semantic quality requirements.** Automatic control of these requirements is very difficult, to some extent probably unsolvable. Nevertheless some methods for automatic analysis, mainly originating in the information retrieval field, are available today. So, the language of a document is recognizable or – more complex – similarities between neighboring documents can be measured. If a hyperlink points to a document too dissimilar to the anchor document, the editor will be warned, because the coherence of the contents might be reduced. Analysis of contents can also help in retrieving documents [5].

2.3 Editorial Group Work

As mentioned, online-publishing group work presents the problem of providing a consistent view to external information. While direct access to the network by the group members would lead to inconsistent views as soon as a remote document accessed twice is changed (or has become inaccessible), a central network cache mediating the network access of the group could provide a consistent even though not necessarily up-to-date view. As a side effect, this cache allows efficient network use, resulting in decreasing network costs.

2.4 Online Publishing: the Model

Fig. 1 shows our model of the online publishing process structuring it into three steps: acquisition (including authoring), editorial processing and publication. The model shows the two information pools the publishing process is based on: a database for management of the editorial group's work and the external information accessible online. Since online information might be imported into the editorial database and editorial information can be exported to the network by allowing online access to these documents held in the database, the model contains a *feedback loop between publication and acquisition.*

3 An Example for an Online Journal: the MultiMedia Forum

The MMF [13] is an electronic online journal in use as an internal company journal at GMD's Integrated Publication and Information System Institute (IPSI). The system is based on a SGML database [1] [8] managing documents according to an especially developed SGML Document Type Definition. Two types of environments

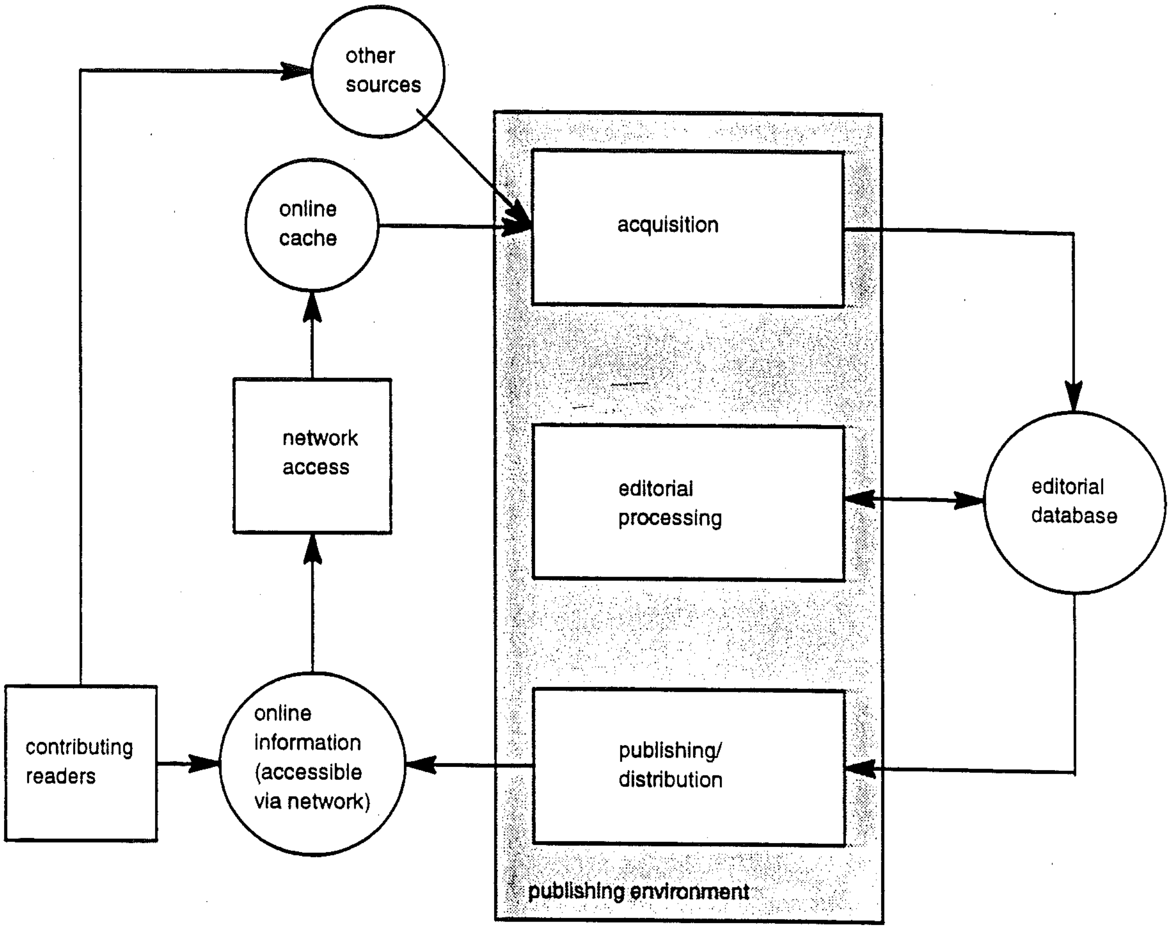


Fig. 1: Online publishing model

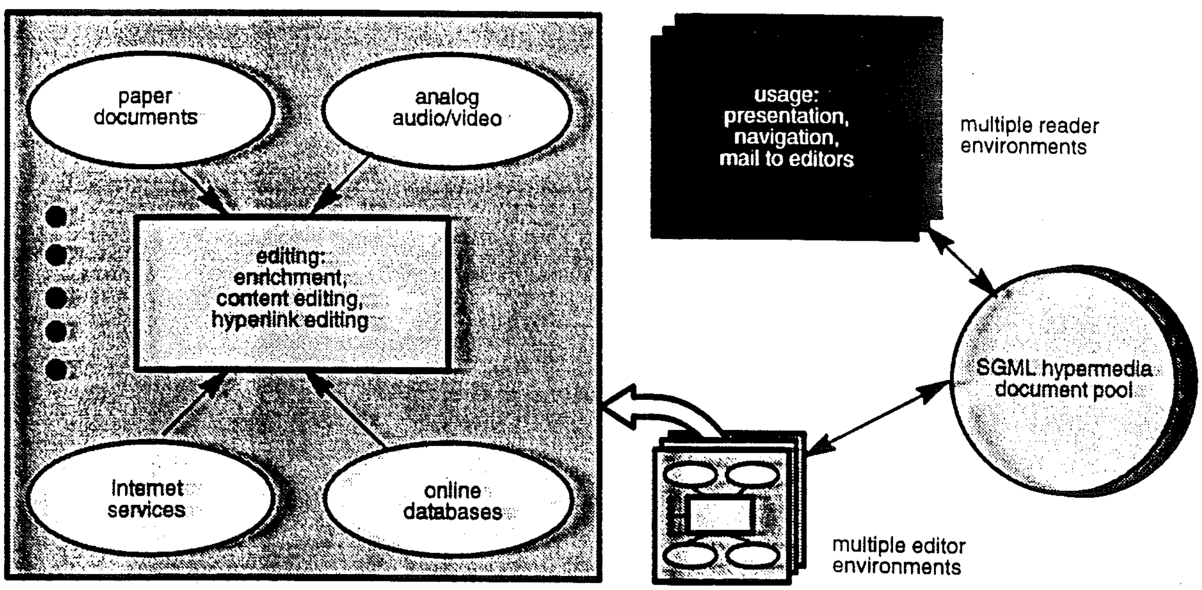


Fig. 2: MMF system structure

– one for the readers, one for the editors – allow access to this database (Fig. 2). Multimedia documents like digital video sequences or images have the same general form as text documents, but contain a reference to a multimedia content file instead of marked-up text content. The textual parts of the documents can contain hyperlinks to other documents of the database – these are addressed by their unique database identifier – or to arbitrary documents in the W^3 identified by an URL. Vice versa the MMF database can be accessed via W^3. The MMF editorial environment is the target system for the BWON system described as follows.

4 The BWON System

4.1 The BWON Architecture

BWON provides special browsing support for editorial work in the MMF editorial environment with online access to the W^3. Because the system focuses on the analysis of links *between* documents and ignores local links connecting parts of *one* document, we will use the notions of "document" and "node" synonymously.

Built as a distributed architecture (Fig. 3), the system's distribution can be described in two ways: Firstly, in BWON different actions are executed in different processes, thus keeping the system from blocking. Secondly, the interface between user interface processes and network access is structured in a client-server fashion allowing coordination of several users. Probably the most important action to be performed is to obtain information. While accesses to the MMF data simply can be done by sending a query to the database system, the

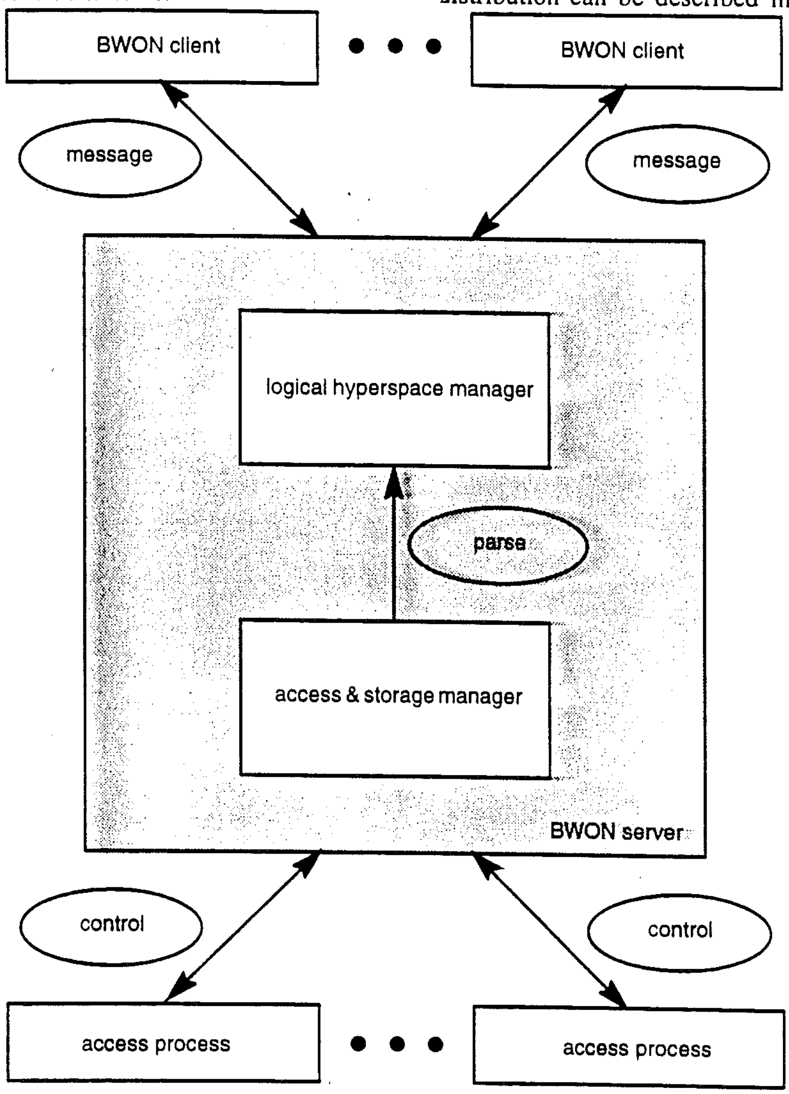


Fig. 3: BWON system architecture

access to the W^3 can take a lot of time and may even be unsuccessful. To smooth these differences we have built the **BWON server** managing accesses to both MMF and W^3 information. The server delegates network operations to **access processes** working independently from each other, controls these processes and caches the results of their work. This caching combined with an appropriate strategy can decrease the number of network accesses, thus reducing costs. The server offers requests to the applications – the **BWON clients** – resulting in such subnets of the MMF/W^3 the applications can use.

The BWON server is structured into two modules, the **access & storage manager** and the **logical hyperspace manager**. The first module arranges the access to requested documents and keeps complete copies of them. The second module contains a data structure which represents the hyperlink topology of all nodes obtained so far and some selected information about the nodes (node addresses, titles etc.)[1]. A parser was implemented to extract the logical structures from the document copies. Furthermore, the modules handle the communication with other processes of the system. The access & storage handler decides whether to search for MMF or W^3 data, creates appropriate access processes and hides the difference between both access nodes from the logical hyperspace manager. Since each access process is responsible for exactly one document to fetch, multiple accesses can be managed easily.

The logical hyperspace manager handles the communication with the BWON clients. On the one hand this means responding to client requests like querying for a list of a document's links or for properties of a node. On the other hand the server asynchronously sends update messages to the clients as soon as a previously retrieved part of the hyperweb has changed its topology. This makes a consistent view for all clients possible.

4.2 The BWON client application

The BWON client we built for editorial support is the only process in the BWON architecture which has a user interface (Fig. 4). Depending on user actions it fulfills two tasks:

1. It shows an interactive mapping of the W^3/MMF hypergraph browsed so far. This will help the user to recognize relations between documents to be published and other nodes of the W^3 or the MMF (see chapter 2).
2. The client enables the editor to apply functions of the MMF editorial system to nodes of the graph, e.g. document presentation (using Mosaic for W^3 contents and the MMF document-viewing facility for MMF documents), document editing (only for MMF documents), conversion from HTML to MMF etc.

1. To avoid confusion we would like to point out that the *storage layer* of the Dexter Reference Model [7] corresponds with BWON's logical hyperspace manager.

Because the functionalities of the MMF are out of this paper's scope, in the following solely the first task will be dealt with in some depth.

Although many different modes of interaction and navigation behavior exist for hypermedia systems [6], none of them can fulfill all requirements a user can have. Therefore the BWON client provides both a redundant user interface presenting the information in graphical and textual form simultaneously and the possibility to select a graph layout method. The offered graph algorithms have in common that they insert nodes to be added due to a user action without changing the graph, unless the user requests this explicitly. All graph displaying is based on small icons representing the nodes. By variation of colour and displayed symbols the icons give some information about the node.

The last node the user has entered plays the role of the **focus node** to which functions like presentation or editing can be applied. Its document title is shown as a label on the left side of the window atop a list of its neighbour nodes. Below that a history list of all previously visited focus nodes is displayed. The items of both lists are active, i. e. a mouse click on an item will change the focus node. The graph to the right of the lists shows the section of the hyperweb the user has navigated through. The icons have three characteristic attributes. The *background colour* indicates the location of the node: blue nodes are residing in the MMF database, the white ones are W^3 addresses. The *foreground colour* symbolizes its current role in the navigation process: while the default foreground colour is grey, the focus node is drawn red and the nodes directly accessible from it are black. The *images* in the icons represent the type of the document. Presently only icon images for the MMF document types have been defined: text, audio, video, picture or announcement. Additional information about the node under the mouse pointer (you don't have to click to get it) is presented in a separate window to be placed wherever the user wants it to avoid overloading of the user

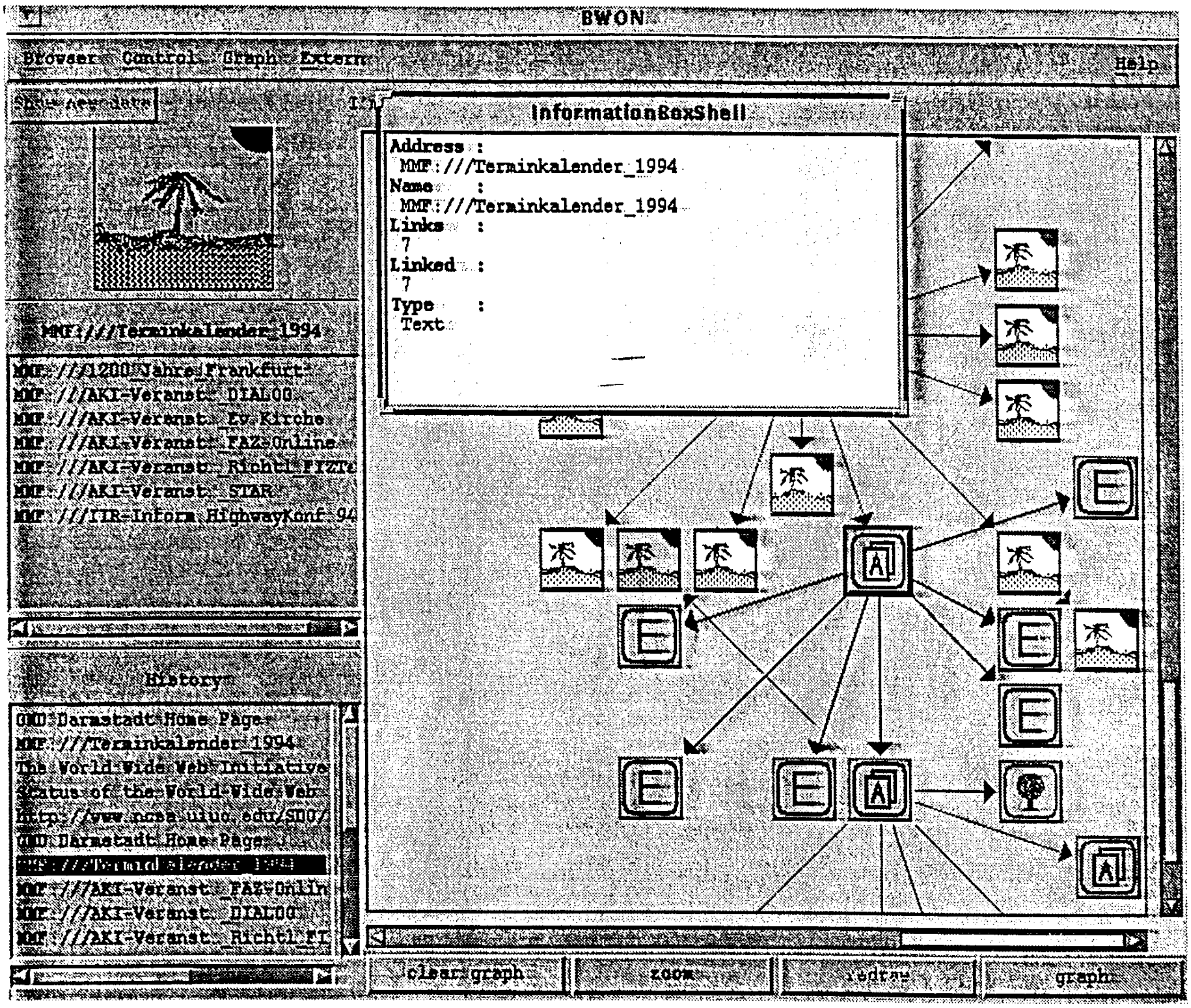


Fig. 4: BWON interface tool

interface. Clicking on an icon changes the focus node, retrieves the list of its link addresses from the server and enhances the graph if needed.

4.3 Filter

Because the number of nodes to be mapped can be too big to be manageable for the user, she can reduce the number of nodes to be displayed by specifying a filter. Furthermore filtering can help to guarantee product quality as mentioned before. There are several ways to get the input for such a filter. The key questions are (a) what information about a node can be obtained and interpreted in an acceptable period of time and (b) what information is needed to fulfill the quality requirements. The transfer of the whole document to the filtering system does not seem to be a good strategy, because this would apply to an unpredictable number of documents. So we distinguish three types of information acquisition. The information about a node can be extracted

1. from the address,
2. from the head elements,
3. from the contents.

The transfer costs of these information are increasing in the listed order. Selecting nodes by their addresses is very simple. Since the address of a node is known in the moment where some link on it is found, additional network retrieving is not necessary. But there are two difficulties in this case: firstly it is not known whether this node is really reachable and secondly the address of a node may have nothing to do with its contents. Therefore examining a node's head elements might be a better

way because these elements describe the node in a uniform way. The title of a document usually gives some idea of its contents and the header's metainformation can help to classify the documents. Unfortunately few W^3 servers are known which offer the transfer of head information separately although HTTP provides this mode. Furthermore since using head information is unusual, authors don't care about it. Most information can be obtained from the document contents. Filtering by contents is most effective but implies the transfer of whole documents.

We define an **elementary filter** by a regular expression representing a set of strings. These filters can be combined by Boolean operators the resulting Boolean expression being a **step filter.** A step filter can be applied to all nodes in a definite distance from the focus node, e. g. the set of nodes reachable from the focus by following exactly two hyperlinks. These filters are applied conforming to the rule, that the step filter for distance n must not be weaker than the step filter for distance n_1, i. e. with growing distance from the focus less nodes are passing the filter. This results in a graph displaying all nodes (or nearly all) near the focus node and only the most important nodes in some distance similar to a fish-eye view.

The filter technology can be used to control the quality requirements (cf. section 2.2). In most cases technical quality requirements can be checked by examining the address or the head information. The control of structural quality requires at least the head elements but for most document formats the whole document contents are necessary to find out the linking structure. Semantic quality control can only be based on the document's contents itself.

5 Conclusions

Concerning the network support for online-publishing applications we are able to draw two conclusions from the experiences made with BWON:

1. Filter technology is very important for the editorial work and thus should have some influence on the network implementation. The protocol(s) and consequently the servers have to support requests for metainformation, e.g. about link structures, document size etc. That is, the document format (in our application, HTML) influences the server implementation.
2. Development of an appropriate cache strategy for the W^3 is a hardly solvable task, because there are no general rules describing the update rate of the documents available by network access. It seems to make more sense to build up a propagation mechanism: clients may subscribe to a server, if interested in document changes on this server. The server replies with messages to the client (in our case the BWON server) containing the updates or at least a hint about the update. This mechanism may also work on a per-document base and will be available at low charge in a commercial system.

The BWON system has been implemented on a UNIX Workstation using OSF-Motif and C++ (at the time of writing, the filter module is not fully implemented) and is used by the editorial staff of the MMF.

Acknowledgements

We owe special thanks to all members of the MIPP department and to Andrea Köhler (student at FH Darmstadt), who has given us some valuable hints for chapter 2.

Literature

[1] K. Aberer, K. Böhm and C. Hüser, 'The Prospects of Publishing Using Advanced Database Concepts', in *Electronic Publishing*, 6(4), 469-480, (December 1993).

[2] T. Berners-Lee, 'Universal Resource Identifiers in WWW', via http from http://info.cern.ch/hypertext/WWW/Adressing/URL/uri-spec.ps, CERN, Geneva, Switzerland, March 12th 1994.

[3] T. Berners-Lee, R. Cailliau, J.-F. Groff, and B. Pollermann, 'World-Wide Web: The Information Universe', in *Electronic Networking: Research, Applications and Policy*, **1**(2), (Spring 1992).

[4] T. Berners-Lee and D. Connoly, 'Hypertext Markup Language. A Representation of Textual Information and Metainformation for Retrieval and Interchange', Internet Draft, July 13, 1993.

[5] P. De Bra et al., 'Information Retrieval in Distributed Hypertexts', in Intelligent Multimedia Information Retrieval Systems and Management, Conference Proceedings RIAO 1994, 481-491.

[6] J. Conklin, 'Hypertext. An Introduction and Survey', in *IEEE Computer*, **20**(9), 17-41, (Sept 1987).

[7] F. Halasz, M. Schwartz , 'The Dexter Hypertext Reference Model', in *Communications of the ACM*, **37**(2), 30-39, (February 1994).

[8] C. Hüser and A. Weber, 'The Individualized Electronic Newspaper: An Application Challenging Hypertext Technology', in *Hypertext und Hypermedien 1992: Konzepte und Anwendungen auf dem Weg in die Praxis*, eds. R. Cordes and N. Streitz, 62-74, Springer, Heidelberg, (1992).

[9] S. R. Newcomb, N. A. Kipp, and V. T. Newcomb, 'The HyTime Hypermedia/Time-based Document Structuring Language', in *Communications of the ACM*, **34**(11), 67-83, (November 1991).

[10] N. J. McCarthy, 'Directing Traffic', in Publish, November 1994, 75-82.

[11] NCSA, *About NCSA Mosaic for the X Window System*, via http from http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/Docs/help-about.h, University of Illinois at Urbana-Champaign, National Center for Supercomputer Applications, 1994.

[12] N. A. Streitz, J. Haake, J. Hannemann, A. Lemke, H. Schütt, W. Schuler and M. Thüring, 'SEPIA: A Cooperative Hypermedia Authoring Environment', in Proc. 4th ACM Conference on Hypertext (ECHT'92), Milano, Italy, Nov. 30 - Dec. 4, 1992, 11-22.

[13] K. Süllow, I. Gabel-Becker, M. Ockenfeld, W. Putz, and G. Roth, 'Multimedia Forum – an Interactive Online Journal', in *Electronic Publishing*, 6(4), 413-422, (December 1993).

Evaluation of a Query Language for Structured Hypermedia Documents

John F. Buford
Distributed Multimedia Systems Lab
Department of Computer Science
University of Massachusetts Lowell
Lowell, MA 01854
buford@cs.uml.edu

Abstract

Structured document query languages have been proposed for providing dynamic linking and composition behavior within hypermedia systems. HyTime (ISO 10744) defines a structured document query language called HyQ, but HyQ has received little attention by designers of hypermedia systems. We discuss and evaluate the use of HyQ for dynamic linking and virtual document structures. A series of HyQ examples are provided as part of the presentation. We also review qualitative performance considerations and implementation status.

Keywords: Distributed hypermedia systems, structure-based query languages, HyTime, HyQ, information retrieval

1. Introduction

The Hypermedia Time-Based Structuring Language (HyTime) [ISO92] defines a query language (HyQ) for dynamically selecting components of a HyTime structured document based on both structural attributes and content. We have implemented a HyTime document database and engine called HyOctane™ [Buford94a], and are currently extending this system to include the HyQ facility. We consider HyQ to be one of the more novel features of HyTime. In this paper, we discuss the use of HyQ for dynamic hypermedia webs and virtual hypermedia structures, facilities proposed by Halasz [Halasz88] in his roadmap for third-generation hypermedia systems. We summarize Halasz' proposal later in this section, and use it as a basis for our analysis of HyQ since much of his proposal is still unachieved by widely used hypermedia systems, and Halasz' framework has been influential in the hypertext research community.

1.1 Structured Hypermedia Documents

HyTime is an extension of SGML in to the domain of hypermedia and multimedia documents. HyTime is a structured document architecture with a rich set of primitives for designing document models. From the HyTime perspective, HTML, a specific hypertext document model used in the World Wide Web (WWW), is one of many possible document models that can be described using HyTime [Rutledge94]. HyTime is based on the SGML notion of document markup. Like SGML, a HyTime document contains markup that identifies the document's logical structure. Use of structure markup exposes semantic and logical structure of the document to the application, augmenting text retrieval facilities and enabling model-based processing of the document. Currently, automated methods for identifying document structure are still a research issues.

We are interested in the use of hypermedia document architectures such as HyTime in distributed hypermedia systems such as WWW [Berners-Lee93] or Hyper-G [Kappe94]. These systems today provide a single document model (HTML and HTF, respectively). Use of a standardized document architecture such as HyTime enables an application to define its own document model in a portable fashion. In [Buford95b] we describe a straight-forward mechanism by which a distributed hypermedia system can provide an open-document model system. In this approach, HTML, HTF and other hypermedia document models are represented as HyTime applications, transparently to the end-user. Using HyTime has several benefits: 1) many of the temporal and multimedia modeling limitations of existing document models can be overcome, in a standardized way, by providing new HyTime-based markup tags; 2) new document models can be defined by the application developer and delivered anywhere on the distributed web without requiring special programming by the system designers; 3) HyQ and other powerful HyTime facilities can be used by applications.

1.2 HyQ and Computational Hypermedia

According to Halasz [Halasz88], there are seven issues for third-generation systems (Table 1). Several of these issues relate to the functionality provided by HyQ, and we summarize them here.

Table 1: Halasz' [Halasz 88] seven issues for third-generation hypermedia systems

Issue	Description
Integration of Search and Query Functionality	Integration of information retrieval and DBMS facilities; pattern matching languages for hypergraph manipulation
Composite Node Types	In addition to content nodes, need container or collection nodes
Virtual Structures over Node Collections	Computationally-defined hypergraphs, analogous to database views
Computation over Hypermedia Networks	Integrated computational engines available to the application
Versioning of Nodes and Subgraphs	Maintaining change history at both node and graph level; version identification as an attribute in search and query
Support for Collaborative Work	Support for shared access to hypertext, and group protocols
Extensibility and Tailorability	Easier customization of hypermedia system by end user

Issue 1: Search and Query in a Hypermedia Network. Halasz distinguishes between content search (traditional information retrieval) and structure search. For Halasz, structure search deals with the node and link semantics as opposed to node content semantics. Using structure search, queries such as the following can be expressed [ibid., p. 843]:

> "Find all subnetworks containing an 'Issue' node linked to at least two 'Position' nodes, each of which has no outgoing links."

Later in the paper we show a HyQ instance for this query. Halasz suggests one use of such a query facility is to customize the view of hypermedia browsers.

Issue 3: Virtual Structures for Dealing with Changing Information. One problem experienced by users of NoteCards was that of premature organization [ibid., p. 845], that is, the tendency of users to create a static hypertext link and node structure before the information structure is fully understood. This problem is a result of the static nature of most hypermedia networks. Any changes to the organization of the node and link structure have to be manually performed in today's systems. A virtual node is one whose contents could be determined dynamically whenever the node was accessed, e.g., "a subnetwork containing all nodes created by someone other than me in the last three days." Similarly, links might be determined dynamically, e.g., "one could link from the ClaimX node to the node containing the currently strongest evidence that supports ClaimX."

Issue 4: Computation in (over) Hypermedia Networks. Integration of a computational engine with a hypermedia storage system enables more powerful, dynamic delivery of hypermedia content. Many systems today provide scripting languages to achieve some of this functionality. As discussed later, HyQ is a query language for structure-based retrieval. The use of scripting languages and other computational engines in hypermedia systems seems to be a separate issue, although HyQ could be used by a computational engine.

Because of HyQ's specific scope for structure-based retrieval, certain operations needed for dynamic hypermedia are not available in HyQ. Currently HyTime does not provide for scripting language facilities [Buford94b].

In summary, Halasz' prescription for third generation systems proposes a tight coupling between computational facilities and the hypermedia document structure. HyQ is a query language for the HyTime hypermedia document model, an international standard, and is the only structured document query language we are aware of. No analysis or implementation of HyQ has been given yet in the research literature. The specification of HyQ in ISO 10744 provides few examples of its use. We present a series of example HyQ queries in section three which illustrate some of the capabilities and limitations of the language. Note that direct use of HyQ queries is not something that an end-user would likely be exposed to. It has been proposed [DeRose 94] that a visual programming interface would be more appropriate for end user queries. Since this paper is concerned with evaluating the capability of HyQ, we show queries directly in the HyQ syntax.

In section four we consider the application of HyQ to problems similar to those suggested by Halasz. The remainder of the paper discusses performance issues and implementation status. The next section gives an overview of HyQ and its relation to scripting languages and information retrieval methods.

Table 2: HyTime location addressing forms used in HyQ

HyQ Operator	Type	Description	Example
Proploc	Property	Locate object by property	Proploc(CAND ATTNAME)
Listloc	List	Locate object by position in list of objects	Listloc(CAND (1 3))
Treeloc	Tree	Locate object by its position in the document tree	Treeloc(CAND 1 2 3)
Pathloc	Path	Locate object by a path index in the document tree plus a position in a path	Pathloc(CAND (1 2))
Relloc	Relative	Locate object by its position in tree relative to other object	Relloc(CAND root anc (1 1))
Dataloc	Data	Locate object by coordinate addressing	Dataloc(CAND (3 4))

2. HyQ: A Query Language for HyTime Documents

2.1 Overview

HyQ is a query language for HyTime documents. Using the structure of a document defined by its SGML/HyTime document type definition (DTD), document elements can be queried by attribute, properties, data content, or relation to other document elements. For example, a simple HyQ query to identify all citation nodes in a document with id "thisdoc" might be written:

```
<HyQ qdomain=thisdoc>
  Select(DOMTREE
      EQ(Proploc(CAND GI) "citation"))
</HyQ>
```

This query returns each element in the document with the property that its general identifier is the token "citation". The result of a query can be used in several ways in HyTime, e.g.: 1) as an endpoint of a hyperlink, 2) as embedded content in another document element. Queries are not restricted to a single document file.

HyQ is a restricted functional language which operates on document objects called node lists. In SGML there are four nodes types: elements, pseudo elements, data entities, and data content. The examples in this paper will deal only with element nodes. Node lists are ordered according to their position in the document tree of the SGML document. All HyQ operations preserve this order.

HyQ's select operator resembles the SQL select function. A node list can be treated as a set, and the basic functions union, intersection, difference are provided. New node lists can be formed by selection from another node list. Selection can be based on node attributes, properties, relation to other nodes, or a match on an element's data content.

Object referencing is a key facility in HyTime referred to as location addressing. There are seventeen architectural forms in the HyTime location address module. Location addressing in HyTime is a generalization of the concept of anchors in hypertext. HyTime location addressing builds on the separation of links and anchor specifiers as found in the Dexter Hypertext Model. Additionally, location addresses can be chained (forming location ladders). HyQ permits certain location address constructs to employed directly in HyQ queries. Table 1 lists these forms.

HyQ uses the HyTime lexical notation (HyLex) for matching data content within nodes. HyLex is equivalent to regular expressions. An application can also define its own lexical notations.

The grammar for HyQ and a brief description are given in an appendix of [ISO8859-92] Other references include [Kimber92, DeRose94].

2.2 Relation of HyQ to Scripting Languages

Scripting languages have become popular in hypertext and multimedia authoring tools. Such languages are typically used to specify the interactive behavior of hypermedia applications and can operate on the presentation structure and content dynamically. Furthermore, scripting languages provide a large set of functions for accessing external services such as the operating system, databases, GUIs, etc. Most systems

which use scripting languages have a weak notion of document structure.

In comparison, HyQ has few of the features of scripting languages, but is tightly coupled to the SGML/HyTime structured document model. Document query languages seem to be special cases of data manipulation languages (DMLs) for object-oriented databases, such as that defined in [ODMG93]. From an object-oriented data model perspective, HyQ fixes the set of classes and their relations for the semantics expressed in the SGML/HyTime document architecture. HyQ does not provide operations for modifying document objects or their attributes as is found in DMLs and scripting languages.

One consequence of the synergy between HyQ and object-oriented data manipulation languages (DMLs) is that it is natural to use an object-oriented database interface to implement HyQ, as we discuss in a later section.

The use of scripting languages with HyTime is discussed in [Buford94a].

2.3 HyQ and Information Retrieval

Common information retrieval methods like boolean and vector space do not distinguish structural aspects of documents. Query matches are typically ranked in terms of relevance. As a content retrieval method HyQ's match function is primitive in comparison. Structure-based retrieval can express relationships that are not expressible in typically IR methods. HyQ should be seen as complementary to common IR techniques.

3. Examples

3.1 Using Select

The Select function produces a node list whose members all satisfy some test made against the input node list. For example, we might select only those nodes which have a specific attribute, as shown in Example 1. A common use of Select is to combine it with a test expression based on property location addressing. Example 1 uses the proploc addressing mode to test the ATTNAME property of nodes in the input domain.

Example 1. Selecting by ATTNAME property

```
<!-- list of nodes which have given
attribute -->
<HyQ fn=hasatt>
 Select(DOMTREE EQ
    (Proploc(CAND ATTNAME ignore)
%1))
</HyQ>
```

The structure of this query is similar to others that use the Select function. The first argument of Select is the node list. In this example, the keyword DOMTREE indicates that the domain of the query is to be the entire tree descended from the input domain. If DOMROOT were used instead, the query would test only the root node in the input domain. The second argument to the Select function is a boolean expression which must evaluate to true for any node that is selected. This expression is the selection criteria.

In example 1, the selection tests each node, as specified by the candidate (CAND) operator. The test succeeds if the node has an attribute name equal to the argument passed when the function hasatt is used. The query is defined as a function so that it can be used in other queries. It takes one argument, the attribute name used to select nodes, and this argument replaces the parameter marked by "%1". User-defined HyQ functions can take any number of arguments, each one appearing in the body of the function according to its position in the argument list, e.g., %1, %2, etc.

The use of proploc in a HyQ expression is equivalent to the proploc ETF. The syntax of proploc and other location addressing forms in HyQ is given in example 2, where optional parameters are marked by the '?' operator. All location addressing forms can be used in HyQ queries except notloc, nameloc, and bibloc. The spanloc ALF and portions of the multloc ALF do not apply to these forms when used in HyQ. The use of the ALF locsrc is represented by the node list argument in the first position of the operators in Example 2. Only the set attribute in the multloc ALF applies to listloc, treeloc, pathloc and relloc.

Example 2. Location addressing forms available in HyQ

```
proploc( (node|nl) qpn joint?
         appropsrc? notprop? )
listloc( nl mkpair* overrun? set? )
treeloc( nl snzi* overrun?
         set? treecom? )
pathloc( nl mkpair*
         overrun? set? treecom? )
relloc( nl (node|nl) relation?
      mkpair* overrun? set? )
dataloc( nl quantum? catsrc? catres?
      mkpair* overrun? )
```

The next example uses the hasatt query function defined in example 1 to create the function hasatnv. This function selects nodes which have an attribute with a given name and value. It uses hasatt to first select all the nodes with matching attribute name from the input domain. The result of this function is then the domain for the second Select, which checks each node for a

matching attribute value. The user-defined function, hasatt, is invoked using the UseQ function, which makes the indirect query to hasatt and passes the arguments as well.

Example 3. Selecting by ATTNAME and ATTVAL properties

```
<!--list of nodes which have
attribute name and value -->
<HyQ fn=hasatnv>
 Select(UseQ(hasatt DOMTREE %1)
    EQ(Proploc(CAND ATTVAL) %2) )
</HyQ>
```

The behavior of a nested function can also be seen by replacing the function name with the body of the function. Example 4 is equivalent to example 3 with this substitution. Evaluation of nested functions always proceeds from innermost to outermost.

Example 4. Expanding the function in example 3

```
<HyQ fn=hasatnv>
 Select(
    Select(DOMTREE EQ(
       Proploc(CAND ATTNAME ignore)
%1))
    EQ(Proploc(CAND ATTVAL) %2) )
</HyQ>
```

Some properties have specifiers which can be used to restrict the domain of a given attribute. Rather than nesting queries as shown above, a specifier can be included as shown in example 5, further simplifying the definition of the function and potentially increasing its efficiency. In this example, ATTVAL is a specifier for ATTNAME.

Example 5. Using specifiers.

```
<HyQ> fn=hasatnv>
    Select(DOMTREE EQ(
       Proploc(CAND ATTNAME[%1]) %2))
</HyQ>
```

The previous examples define HyQ functions which can then be used in a HyQ query. Two sample queries are shown in example 6. The first instance uses attributes to name the function and the arguments to be passed to it. The second instance uses the UseQ operator to invoke the function. These two forms are equivalent.

Example 6. Using previously defined functions--two cases.

```
<HyQ domain="thisdoc"
    usefn=hasatnv args="ID citation">
<HyQ domain="thisdoc">
    UseQ(hasatnv ID citation)
</HyQ>
```

Select, the cornerstone of HyQ, is straightforward to use as a nodelist filter when the assertion is a test against a property or attribute of nodes in the nodelist. In this case the assertion contains the CAND symbol, which causes each node in the list to be substituted in the assertion expression where the CAND occurs. If, instead, filtering of nodes is needed based upon an object value external to the nodelist, the assertion needs more complex formulation. For example, suppose we wish to select between node A and B depending on the value of property X for node C (e.g., select A if C has the right property X value, otherwise select B). The following example doesn't work, because it produces either (A B) or () instead of A or B:

Example 7. Trying to select between node A or B based on the value of node C's property X.

```
<HyQ>
    Select((A B) EQ(
       Proploc(C X) propval))
</HyQ>
```

Example 8 shows a way to perform this type of selection by rewriting the assertion to be the conjunction "if node is an A node and property X holds for node C, or if node is a B node and property X doesn't hold for C".

Example 8. Filtering based on conditions external to the nodelist being filtered.

```
<HyQ>
    Select((A B) Or(
      And(EQ(CAND A) EQ(
         Proploc(C X) propval))
      And(EQ(CAND B) NE(
         Proploc(C X) propval))
      ))
</HyQ>
```

When A and B are nodelists rather than single nodes, then the equality test in example 8 must be replaced with a nested selection as shown in example 9. Notice that CAND is used in more than one context. The ambiguity is resolved by the rule that CAND refers to the innermost select.

Example 9. Extending example 8 when A and B are nodelists instead of single nodes.

```
<HyQ>
 Select((A B) Or(
    <!--needed for ref in
         inner Select-->
    Assign(temp CAND)
    And(Select(A EQ(CAND
              Nlref(temp)))
         EQ(Proploc(C X) propval))
    And(Select(B EQ(CAND
              Nlref(temp)))
         EQ(Proploc(C X) propval)) ))
</HyQ>
```

3.2 Selecting Nodes by Relative Position

Using relative location addressing, it is possible to locate nodes by relative position in the document tree. Possible relative positions include parent, child, sibling, ancestor, and descendant. The following example uses relative addressing to find the parent node, given an arbitrary node in the document.

Example 10. Finding the parent of a node

```
<!--parent of this object-->
<!--sibling, child, etc. can be done
similarly-->
<!--%1 current node, %2 root node-->
<HyQ fn=parent>
 Relloc(%1 %2 relation=parent
overrun=trunc)
</HyQ>
```

Relloc requires the root node of the document as a parameter. This can be found using the "docroot" property which is maintained for every element. The following example finds the root node of a document, given an arbitrary node in the document using the docroot property.

Example 11. Finding the root node of a document from a given node.

```
<!-- find the docroot, %1 is an
    arbitrary element in the document-
    ->
<HyQ fn=docroot>
     Proploc(%1 docroot ignore))
</HyQ>
```

3.3 Node List Operations

A node list is ordered according to position of the node in the document tree, as encountered by a depth-first traversal of the tree. Node lists can be produced which duplicate a node. Example 12 shows two ways to do this. Which duplicate is removed? Presumably the first in the list is retained, and any repeated nodes are removed as encountered. The standard doesn't clarify this.

Example 12. Removing duplicates in a nodelist

```
<!--remove dup nodes in nodelist-->
<HyQ fn=remdup>
  <!--one way to remove duplicates-->
  Listloc(%1 <!--mkpair=-->
             (1 -1) set)
</HyQ>

<!--remove dup nodes in nodelist-->
<HyQ fn=remdup>
 <!--another way to remove dups-->
  Union(%1)
</HyQ>
```

3.4 Iteration and Recursion

The Select operator allows individual tests against all nodes in a nodelist, so it is effectively a composite iteration function. It is designed to be used as a filter, in which the assertion argument specifies a condition which any input node must satisfy in order to be included in the output node list produced by select. HyQ has a restricted use of state information (the Assign operator can only be used once per name in a given context). And there is no straightforward way to express conditional execution of a previously defined function. We next explore these issues by developing a query to find the minimum or maximum of some attribute of a node list. For example, the following queries might be of interest to an application:

Find the node with the largest number of children
Find the node with the greatest number of attached links
Find the node with the attribute "modification date" which has the most recent value
etc.

In order to find the maximum or minimum, the currently known minimum or maximum must be accessible and revisable at each step in the iteration over the node list. The next example does this, but violates the constraint on the HyQ operator Assign. It also relies on the And operator being sequential (i.e., evaluate operands from left to right until the first False condition is tested), which isn't discussed in the standard.

Example 13. An incorrect way to find the node with a maximum of some attribute.

```
<HyQ>
 <!--the current max is initially the
    root node-->
 Assign(maxnode DOMROOT)
 Select( DOMTREE
  And( <!--assume sequential And,
    which isn't stated by standard-->
  <!-- this trick, sort of a
       conditional statement,
 works only if AND evaluates members
 until not true one is found -->
 <!--application's test function
    for max of some attribute-->
 UseQ(test CAND Nlref(maxnode))
 <!--can't redefine name within scope
    of single query
       so this doesn't work ... -->
 Exists (Assign( maxnode CAND)) ))
       Nlref(maxnode)
</HyQ>
```

The next example uses a recursive approach instead[1]. The trick here is how to terminate a recursive procedure. Example 14 builds the termination condition as a dynamic node list. When the termination condition is reached, the node list will contain the "stop" node; otherwise, the node will contain the recursive call.

Example 14. Find max node using recursion.

```
<!-- sample comparison function -->
<HyQ fn=test>
 Listloc(
    Union(
      Select( %1 GT(%1 %2))
      %2)
 1 1)
</HyQ>

<!--recursion termination function,
  returns empty node list-->
<HyQ fn=stop>
    Create()
</HyQ>

<!--max function-->
<HyQ fn=max>
 <!--head is first elem in the nl-->
 Assign(head Listloc(%1 1 1))
    <!--termination test: use
      different fn when head empty
      func is "stop" when head empty,
      "max" otherwise.-->
 Assign(func
    Listloc(Union(stop max
      Nlref(head))

    2 -1))
 <!--tailmax is max elem in nl tail--
 >
 Assign(tailmax
    Query(Nlref(func)
      Listloc(%1 2 -1)
      Listloc(%1 2 1)))
 <!--Obtain the answer from the
      dynamically formed list-->
 Listloc(
    <!--if select condition
 succeeds,
         the first node in the
 nodelist
         is tailmax, otherwise
         the first node in the
 nodelist
         is head-->
    Union(Select( tailmax
         And(tailmax
           Useq(test head tailmax)))
         head)
         1 1)
</HyQ>
```

A third approach to finding maximum or minimum value of an attribute or property of a node list is to position each node on an finite coordinate space axis using the scheduling module, based on the given attribute or property. Given this mapping, it should be possible to find minimum and maximum positions in a range. However, such a construction appears awkward and inefficient in general, although it would be useful if such a construction were needed for other purposes.

In summary, what should be a relatively simple query--finding the minimum or maximum of an attribute in a node list--requires a convoluted construction in HyQ because of the lack of variable state and conditional expressions outside of Select operations. Evaluation order is not clearly specified for And and Or operators. The examples given in this section, though demonstrating some possible programming techniques, seem to stretch the language.

3.5 Sorting Nodelists

HyQ operations always preserve node list order. In general this seems desirable. However, when using HyQ to construct dynamic views, it seems important to be able to reorder nodelists according to some attribute or value. A built-in sort operator seems like a good feature to add to HyQ.

3.6 Creating Nodes Dynamically

Most of HyQ's functionality has to do with creating nodes lists selected from nodes in the document tree. An application might also want to create document structure dynamically. There is a create nodelist function (example 15). There doesn't seem to be a way to parameterize the delimited data so that a new node could be constructed computationally.

Example 15. Create a new node from delimited data--two examples.

```
<HyQ fn=newlink>
 Create("<lnk anch=etc></lnk>")
</HyQ>

<!--Create a new elements-->
<HyQ fn=anewel>
 Create(%1)
</HyQ>
```

1. The HyQ specification doesn't permit a function to directly call itself ("... fn attribute of *another* query element") but this can be sidestepped by renaming, where recurs1 and recurs2 are otherwise identical:
<HyQ fn=recurs1>UseQ(recurs2 ...)</HyQ> and <HyQ fn=recurs2>UseQ(recurs1 ...)</HyQ>
Consequently, since such cyclic invocations aren't prohibited, recursion is possible. The examples in the text avoid this renaming for the sake of brevity, but it would be required to conform to HyQ syntax.

```
<HyQ>
 UseQ(anewel "<newel id=abc></newel>")
</HyQ>
```

4. Querying a Document Web

The HyTime document web is constructed from element types from the hyperlink and location address modules. The hyperlink module has two architectural forms--clink and ilink--which can be used to define an arbitrary range of link types by an application. Most of the location address forms have been introduced in section two, and these forms constitute a variety of ways to construct the anchors for hyperlinks. The ability to manipulate the hyperdocument web structure thus depends upon HyQ's ability to manipulate nodelists related to hyperlink and location address elements.

Examples of useful dynamic manipulation of the hyperdocument web include:

- Filtering unwanted anchors from the linkend of a link
- Providing a dynamic guided-tour selection of links/ anchors based upon keyword search in which keywords are associated with each node in the web
- Providing a dynamic guided-tour selection of links/ anchors based upon keywords in the content of nodes
- Selecting links by type or attribute for display in a browser
- Identifying composite document structure by identifying patterns in combinations of document types, document root types, and interconnecting link types
- Forming hierarchical web index documents dynamically, so that as new documents are added to the web, the index documents are updated automatically
- Supporting node traversal access policies when node access rights are stored as attributes of a node
- Identifying documents which are within N link steps from the current one

4.1 Using Proploc with Link Properties

Several node properties are available which can be used with HyQ when dynamically manipulating the hyperlink web formed by link and location address architectural forms. These are shown in Table 3, including the brief descriptions given in the standard.

The first example (example 16) returns a nodelist containing all anchors for a given link, using the anchors property for that link. We assume that the anchors are ordered by their order in the anchrole attribute of the ilink architectural form, and that aggregate anchors are ordered within an anchrole by position in the location ladder that connects the aggregate. This nodelist could be used by an application to identify all anchors which are to be highlighted. Or a filtered version of this list might be used as an endpoint of another hyperlink.

Table 3: Link properties (uppercase/lowercase as given in standard)

Name	Description in ISO 10744
hylink	ilink or clink element
anchors	objects linked by hylink
ANCHROLE	anchor roles of anchors of hylink
linkedby	hylinks that link an anchor
linkedto	other anchors of linkedby
LINKEDAS	role of anchors in hylink

Example 16. Obtaining the anchors of a link.

```
<!--list of anchors which this link
   connects-->
<!--assume that the HyPD property
   means
     the endpoints and not the
     anchor objects themselves-->
<!--HyPD doesn't specify the order of
   the objects or their relationship
   to the order of the anchroles in
   the ilink-->
<HyQ fn=anchors>
  <!--anchors: "objects linked by a
      hylink"-->
  Proploc(%1 anchors ignore)
</HyQ>
```

The second example returns all the link nodes which are attached to a given anchor. These links must apparently be those in the same document as the anchor, since it isn't clear how nodes from multiple documents are represented in a node list.

Example 17. Obtaining the links attached to a given anchor.

```
<!--list of link nodes which are
   attached to this object are these
   in the same document or any
   document?

     how are nodes outside the qdomain
     represented in the nodelist? -->
<HyQ fn=linkedby>
  <!--linkedby: "hylinks that link an
      anchor"-->
  Proploc(%1 linkedby ignore)
</HyQ>
```

The next example finds the anchors which correspond to a specific role in the hyperlink (each type of endpoint of a hyperlink is called an anchor role).

Example 18. Finding the anchors for a given anchor role for the hyperlink.

```
<!--anchors by role: anchors for a
   specific end of a link-->
<!--%1 link element, %2 anchor role --
   >
<HyQ fn=anchbyrl>
 Select(UseQ(anchors %1)
    EQ(Proploc(
      CAND anchrole ignore) %2))
</HyQ>
```

4.2 Finding the Web

A browser application might wish to display the documents reachable within one traversal from the current document. The following example finds the document root nodes of the adjacent documents in four steps. Here we assume that properties linked-to and linked-by include objects in other documents.

Example 19. Finding the depth one web from current document.

```
<HyQ>
 <!--1. find all anchors in this doc
    from the links in this doc-->
 Assign(anchors
 <!--use intersection with DOMTREE
    since must be anchors in
    this document-->
 Inter(DOMTREE UseQ(anchors
    <-! find all link nodes -->
    Select(DOMTREE Proploc(
      CAND hylink)) )))
    <!--2. find all the links
        attached to these anchors-->
    Assign(linkedby
      UseQ(remdup
      (UseQ(linkedby
        Nlref(anchors))))
    <!--3. find all the link-to's
      anchors of the linked-by's -->
    Assign(linkedto
      Proploc(Nlref(linkedby)
        linkedto))
    <!--4. return the corresponding
        docroots-->
    UseQ(remdup(UseQ(docroot
        linkedto))))
</HyQ>
```

This query could be used repeatedly in order to extend the web nodelist to more than one level.

4.3 Issue-Position Node Query

Here we show a HyQ query for the example discussed in section 1: "Find all subnetworks containing an 'Issue' node linked to at least two 'Position' nodes, each of which has no outgoing links." The following HyQ solution is performed over a single document, but could be extended to handle inter-document cases.

Example 20. Find all subnetworks containing an "Issue" node linked to at least two "Position" nodes, each of which has no outgoing links.

```
<!--does node have outgoing links?-->
<HyQ fn=hasoutlk>
 <!--1. find all anchors in this doc
    from the links in this doc-->
 Assign(anchors
 <!--use intersection with DOMTREE
    since must be anchors in
    this document-->
 Inter(DOMTREE UseQ(anchors
 <-! find all link nodes -->
 Select(DOMTREE Proploc(
       CAND hylink)) )))
 <!--2. find all the links attached
    to these anchors-->
 Assign(linkedby UseQ(remdup
    (UseQ(linkedby Nlref(anchors))))
 <!--3. find all of linkedby which
    are "outgoing"-->
 Assign(outgo Select(
    linkedby Or(
      EQ(Proploc(CAND ATTVAL[extra])
            "E")
      EQ(Proploc(CAND ATTVAL[extra])
            "A")
      EQ(Proploc(CAND ATTVAL[intra])
            "E")
      EQ(Proploc(CAND ATTVAL[intra])
            "A"))))
 <!--4. is the given node an anchor
 for one of these outgoing links?-->
 Inter(%1 Proploc(outgo anchors))
</HyQ>
<HyQ>
 <!--1. find all issue nodes-->
 Assign(issue UseQ(hasgi
      DOMTREE issue))
 <!--2.find position nodes which
     have no outgoing links-->
 Assign(positn
     Select(UseQ(
         hasgi DOMTREE position)
       Not(UseQ(hasoutlk CAND)))
 <!--3. find the issue nodes which
     have only these nodes as links
     and which have two or more -->
 Select(issue And(
    <!--find attached position
          nodes-->
     Assign(thepos
       UseQ(hasgi UseQ(linkto CAND)
```

```
      position))
   <!--are there any that are not
      in the no-outgoing list?-->
      NE(Inter(thepos positn)
              thepos)
   <!--is the number greater or
      equal to 2?-->
   GE(count(thepos) 2) ))
</HyQ>
```

4.4 Discussion

In addition to these examples we have constructed others which can be used to build dynamically generated guided-tours and dynamic index structures. We thus conclude that it is possible to use HyQ to do sophisticated structure query against the hyperlink web. However, the interpretation of certain hyperlink properties in a multiple document context is not clear.

5. Evaluation

Halasz' "structure search" differs from HyQ because link and anchor structure is only part of the structure of a HyTime/SGML document. Halasz proposed a pattern matching language with regular-expression operators; HyQ is closer to a database query language. There is a match operator in HyQ, but its use is for node content as opposed to the markup structure or node properties.

HyQ is described as a functional language by [DeRose94], one of the principal architects of HyQ. It is a side-effect free language (except for the fairly restricted Assign operator). It has limited conditional execution constructs. It focuses mostly on manipulation by querying as opposed to manipulation by dynamic creation. It appears to permit recursive functions to be written (see footnote in section 3), but because of the lack of conditional flow of control constructs, writing a termination condition for a recursive HyQ function appears tricky. We noted in section three that it is clumsy at best to write HyQ queries which identify nodes which minimize or maximize some attribute. We noted also that the inability to re-order node lists, for example by sorting them, seems to be a limitation. There are no arithmetic operators.

Much of HyQ's power comes from its tight integration with HyTime's location addressing facilities, which are themselves quite rich in variety. The content matching capability, based on the HyLex regular expression language, is weak compared to common information retrieval techniques.

Of course there are ways to extend HyQ, as well as define an application-specific query language as an alternative to HyQ. This would limit the portability of the document, defeating one of the primary purposes of standardizing HyTime.

The use of listloc with node lists allows a Lisp style node list access (example 21), but few other features of Lisp can be emulated. We note that DSSSL is currently being balloted for standardization, and uses a Lisp derivative for its facilities. It seems natural to provide this same power as a programming notation in HyTime.

Example 21. Listloc and Lisp

```
<HyQ fn=car>
   Listloc(%1 1 1)
</HyQ>
<HyQ fn=cdr>
   Listloc(%1 2 -1)
</HyQ>
<HyQ fn=eval>
   Query(%1 %2 %3 %4 %5 %6 %7)
</HyQ>
```

6. Performance

Performance issues fall into two categories: 1) how to write efficient HyQ queries, 2) how to efficiently process queries. The observations in this section are qualitative in nature.

6.1 Writing Efficient Queries

Most non-trivial queries involve nested operations, such as nested *Select* operations. The innermost operations will be evaluated first. Thus an obvious strategy is, when nested operations are order independent, to place the most-restrictive queries at the innermost level of the query. That is, if nested operations can be ordered in terms of the size of the resulting nodelist, then nest the operations so that successful layers of operations are less restrictive.

For example, consider the following two queries which are logically equivalent:

Example 22.

```
<HyQ fn=hasatval>
 Select(
    Select(DOMTREE EQ(Proploc(
          CAND ATTVAL) %1))
    EQ(Proploc(CAND ATTNAME)) %2)
</HyQ>

<HyQ fn=hasatval>
 Select(
    Select(DOMTREE EQ(Proploc(
          CAND ATTNAME) %2))
    EQ(Proploc(CAND ATTVAL)) %1)
</HyQ>
```

Suppose a given document has 500 elements with an average of four attributes per element or about 2000 attributes. The first version tests the 2000 attributes for a specific value perhaps returning 5 matches which are then tested for a matching attribute name, a total of 2005 test operations. The second version tests for attribute name first. This can easily be done in 1250 on average since each attribute name is unique in a given element if half the attribute list of each element is searched on average. If the attribute name is used only in a subset of element types, then this number can be reduced further by first finding element nodes which have that attribute name. The 1250 tests will produce a list of at most 500 matches (but probably substantially less than 500). These 500 will then be checked for match by value, so 1750 tests all together. The disparity between these two cases would likely be much larger in general.

The second case places the most restrictive Select statement at the innermost level, making it more efficient in most cases. A HyQ query processor might also be able to reorder such operations to optimize queries.

A third approach is to not nest the select statements, and to use the intersection operator (example 23). This causes the select operator to test the entire domain each time, usually a poor choice unless the domain is needed more than once in the body of the query. The first two assignments result in 4000 tests. The intersection is linear in |set1|+|set2|.

Example 23. Intersecting select results

```
<HyQ>
   Assign(set1 Select(DOMTREE
     EQ(Proploc(CAND ATTNAME) %2))
   Assign(set2 Select(DOMTREE
     EQ(Proploc(CAND ATTVAL) %1))
   Inter(Nlref(set1) Nlref(set2))
</HyQ>
```

6.2 Efficient Processing of Queries

Two observations can be made with respect to efficient processing of queries. A common operation involves conceptually searching over a document tree to locate nodes that fit specific patterns. Searching an entire document tree node by node is one approach, albeit an expensive one. The DTD for each document describes and typically the DTD tree is much smaller than the document instance tree. Consequently, a query evaluation could first consult the DTD to localize the search to specific portions of the document instance. For example, in the above query, the DTD identifies which elements use attributes of a given name. In practice most elements use attributes with different names. By knowing the name of the attribute, only specific nodes in the document tree need to be queried.

From this perspective the DTD tree can be considered an index structure for searching the instance tree for structure-based queries. Storing the DTD tree in the document database is useful for other purposes as well, such as validation of document updates.

The DTD might also be used to reject queries that are inconsistent with the DTD structure, for example, searching for an attribute value which is inconsistent with the lexical type for that attribute.

The second observation is that an efficient dynamic list structure is needed to support nested Select and set operations. Performance optimization can be further broken down into the following areas, each of which can be individually optimized:

1. Using information from the DTD and metaDTD
2. Select
3. Location addressing
4. Set operations
5. Match

Since location ladders (chains of location addresses) can have cycles, a HyQ interpreter must detect such cycles where they might lead to infinite loops.

7. Implementation

We have implemented a HyTime engine that supports a subset of HyTime and uses architectural forms from four of the six HyTime modules. A HyQ parser and query evaluator has been implemented. This is currently being integrated with the HyTime engine, which uses an object-oriented database based on the ODMG-93 specification [Cattell94] for storing document structure.

8. Summary

We are interested in the use of hypermedia document architectures such as HyTime in distributed hypermedia systems such as WWW or Hyper-G. These systems today provide a single document model (HTML and HTF, respectively). The flexibility and power of these systems would be greatly increased by using more sophisticated document architectures as the basis for document models. HyTime is an important candidate both because of its status as an international standard and because of its rich set of primitives. The ISO MHEG specification [Gopal95] is also a possible candidate.

A basic constraint in existing distributed hypermedia systems such as WWW and Hyper-G is that the

document model is fixed. Of course both systems support other document formats, but these other formats are treated as an embedded media type. Although both of these systems use SGML, the generality possible in true SGML systems is not available because, in SGML terminology, the document type definition (DTD) is fixed at the viewer application. If instead an arbitrary DTD were allowed to be used as a hyperdocument model, an arbitrary number of different document structures (types) could become part of a hyperdocument web. The advantage of this approach becomes obvious when one tries to import existing structured documents into a hypermedia system by converting the document into HTML or HTF. Document structure which doesn't map to the target model is lost in the translation. This problem will become more serious when dealing with true multimedia documents that have interactive and embedded dynamic behavior, temporal semantics, etc.

In the current approach, new document model functionality is provided by extending the DTD and the client applications which display this DTD. So, for example, we see proposals for HTML 2.0, HTML 3.0, and so forth. We call this type of evolution of a DTD to encompass more and more functionality the *super-DTD* approach. The evolution of the DTD is in the hands of the system designers, already over-committed with many demands for new functionality. In the *open hyperdocument model* approach, the system designers are no longer the bottleneck in terms of document model innovation, which in any case belongs in the hands of the application developers. While admittedly DTD design is a difficult task, we expect that tools such as HDM [Garzoto 93] will make it possible for application designers to generate custom DTDs interactively, without having to get into the details of SGML coding conventions. Some commercial SGML tools already support DTD design.

Despite all this, even general SGML, although an improvement in terms of generality over the single DTD approach, is not sufficient because of its lack of hypermedia semantics. HyTime, which has an ample set of hypermedia semantics, also allows an arbitrary set of DTDs to be devised.

In [Buford95b] we describe a straight-forward mechanism by which a distributed hypermedia system can provide an open-document model system. In this approach, HTML, HTF and other hypermedia document models are represented as HyTime applications, transparently to the end-user. This approach would make HyQ available to application developers, moving existing systems closer to the functionality of third generation distributed hypermedia systems.

Halasz' framework does not specifically concern a distributed hypermedia environment, but the concepts nevertheless apply. The HyQ language, despite the limitations discussed in this paper, is a useful subset of the functionality needed to achieve the document manipulation envisioned by Halasz. Increased power could be obtained by augmenting HyQ or by integrating procedural and pattern matching languages with the hypermedia document architecture.

9. References

[Berners-Lee93] Berners-Lee, T., and Connolly, D., *Hypertext Markup Language (HTML): A Representation of Textual Information and MetaInformation for Retrieval and Interchange*, http://info.cern.ch/hypertext/WWW/MarkUp/HTML.html, IIIR Working Group, June 1993

[Bertino88] Bertino, E. Query Processing in a Multimedia Document System, *ACM Trans. on Office Information Systems*, Vol. 6, No. 1, !988.

[Buford94a] Buford, J. F., Rutledge, L., Rutledge, J. L., and Keskin, C. HyOctane: A HyTime Engine for an MMIS, *Multimedia Systems*, vol. 1, no. 4, February 1994, pp. 173-185.

[Buford94b] Buford, J.F., Rutledge, L., and Rutledge, J. Integrating Object-Oriented Scripting Languages with HyTime. 1994 Intl IEEE Conf. on Multimedia Computing and Systems, May 1994.

[Buford94c] Buford, J. F., Rutledge, L., and Rutledge, J. Toward Automatic Generation of HyTime Applications. Proc. Eurographics Multimedia 1994.

[Buford95a] Buford, J. F., Gopal, C., and Rutledge, L. Storage Server Requirements for Delivery of Hypermedia Documents. Multimedia Computing and Networking 95. Feb. 1995

[Buford95b] Buford, J. F. A Transfer Protocol for an Open Hyperdocument Model Server. to appear in ED-MEDIA 95, June 1995.

[Cattell94] Cattell, R. (ed) et al. The Object Database Standard: ODMG-93. Morgan Kaufmann 1994.

[Consens89] Consens, M.P., and Mendelzon, A. O. Expressing Structural Hypertext Queries in Graphlog. Proc. Hypertext '89. pp. 269-292.

[DeRose94] DeRose, S. and Durand, D. *HyTime--Making Hypermedia Work*. Kluwer Academic Press. 1994. .

[Garzotto 93] Garzotto, F., Paolini, P., and Schwabe, D. HDM-A Model-Based Approach to Hypertext Application Design. ACM Trans. on Info. Systems (11) 1. Jan 1993 pp. 1-26.

[Gopal95] Gopal, C., and Price, R. Multimedia Information Delivery and the MHEG Standard. Proc. DAGS Elec. Publishing and the Information Superhighway 95, May 1995.

[Halasz88] Halasz, F. Reflections on Notecards: Seven Issues for Next Generation Systems. *CACM (31)7*, July, 1988, pp. 836-855.

[ISO92] ISO/IEC IS 10744. Hypermedia/Time-based Document Structuring Language (HyTime) (August 1992).

[Kappe94] Kappe, F., et al. Hyper-G A New Tool for Distributed Hypermedia. *Proc. Distributed Multimedia Systems and Applications 1994*. Aug. 1994. pp. 209-214.

[Kimber 93] W. E. Kimber. HyTime and SGML--Understanding the HyTime HyQ Query Language 1.1. Aug 1993. Unpublished manuscript.

[Rutledge94] Rutledge, L. HyHTML: A HyTime DTD incorporating the HTML markup model. Posted to comp.text.sgml August 1994.

AUGMENTING TEXT:
Good News on Disasters

Sara Elo
MIT Media Lab
20 Ames Street, Cambridge, MA 02139
elo@media.mit.edu

Abstract

The transition of print media into a digital form allows the tailoring of news for different audiences. This paper presents a new approach to tailoring called *augmenting*. Augmenting makes articles more informative and relevant to the reader. The PLUM system does this by augmenting news on world-wide natural disasters which readers often find remote and irrelevant. Using community profiles, PLUM automatically processes articles to compare the disaster's effects to the reader's home community. The reader, browsing through annotations which PLUM generates, discovers, for example, the scope of the foreign disaster in terms of his community. The reader can also view an article augmented for other communities. By contextualizing disaster articles and making them more informative, PLUM hopes to create a sense of connectedness.

Read All About It...

May, 1995. 7pm in Bellefontaine, a rural town in central Ohio. Dora Newlon, 57, turns on her computer to read the augmented news of the day:

Fig. 1

Augmented Disaster News
Bellefontaine, Logan County, Ohio, United States.

NIAMEY, Niger (Reuter)

Flooding caused by record seasonal rains in the desert state of Niger killed 42 people and left almost 127,000 homeless, the state news agency ANP said Friday.

Heavy rain since July has destroyed nearly 74,100 acres of crops and killed 6,800 animals, the agency said. The worst affected areas are the central Maradi region and the western Dosso and Niamey regions. The losses are estimated at $52 million.

The average annual rainfall of 14 inches has fallen in a single day in some areas. On August 12, national television was forced to halt broadcasts because its studios were flooded.

Niger is located in West Africa, between Algeria and Nigeria. Niger is about six times the size of Ohio.
(Source: World Fact Book and US Census Data)

There are no Nigeriens living in Bellefontaine, nor in Logan County.
(Source: Census Data)

The population of Logan County is 42,000. Therefore, 127,000 people is roughly three times the number of residents of Logan county.
(Source: Census Data)
The total population of Niger is roughly 9 million. This means one person out of 70 has been affected by the flood.
(Source: World Fact Book)

Livestock in Niger is mostly cattle and sheep.
(Source: World Fact Book)

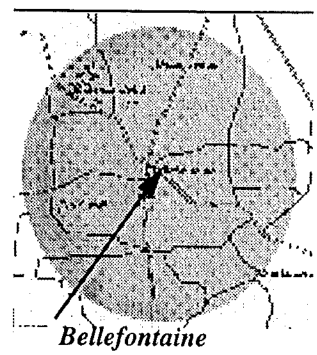
Niger's GNP is roughly $5.4 billion.
(Source: World Fact Book)
To cover an amount equivalent to the cost of the flood in Niger every household in Bellefontaine would have to pay about $1100.
(Source: Census Data)

The last serious flood in Niger occurred in 1991, but yearly floods are normal. The last flood in USA of similar severity was the California flood of 1992.
(Source: CRED Data Base)

The main food crops in Niger are millet, sorghum and rice. None of them are cultivated in Ohio.
(Source: World Fact Book and US Census Data)

9 out of 10 people live from agriculture in Niger. One out of 33 live from it in Logan County
(Source: World Fact Book and US Census Data)

An area roughly equivalent in size to 74100 acres is a circle of radius 6 miles centered on Bellefontaine:



The original article on the Niger flood does not immediately relate to Dora's life in Bellefontaine. Like most of us, she probably knows no one there, where it is, or the difference between Nigeriens and Nigerians.

PLUM uses knowledge about a reader's home community to contextualize news. It explains the disaster in terms of her home town's geography and living conditions. PLUM draws these connections because cognitive scientists recognize that people pay attention to what is familiar. People also understand something new in terms of something they have already understood [Schank 1990].

Motivations for Augmenting Text

Local newspapers often rely on wire services such as Associated Press and Reuters for news outside their community. Small papers like the Bellefontaine Daily cannot afford sending reporters to cover far-away events. From the incoming news wires, editors choose which articles to include in the day's paper. Wire services rarely indicate the relevancy of their articles to their client newspapers' communities. Outside of the obvious references to their state, senator or congressman, the local journalists must research the implications of reported events for their home community. When a highway bill passes the Senate, a journalist uses insight, the local library or other resources to "localize" an article before press time. This is harder with foreign news. When news of the Niger flood arrives, the local journalist must acquaint himself with this distant place and, under deadline, scrambles to find good resources. Given these pressures, smaller newspapers often print international news wires without further refinement for the local readership.

Technology can do more in the newsroom. Most news organizations employ computers to make quantitative improvements, to cut costs, produce faster, and generate better graphics. While 79% of newspapers surveyed by Cable & Broadcasting had computer graphics capability, only 29% had a computerized library and even fewer used information-gathering tools such as CD-ROM databases. [Cable & Broadcasting, 1994]. Technology has yet to significantly improve the quality news.

For example, technology could help counter misconceptions. Foreign disaster news often fosters a tragic image of the developing world. The public has 'an impression that the developing world is exclusively a theater of tragedy, ... This misconception is as profound as it is widespread', said Peter Adamson,[1] author of UNICEF's annual State of the World's Children report [Cate 1993]. Misconceptions arise from ignorance and lack of familiarity.

1. Referring to the 1993 World Vision UK public opinion survey.

Two Bellefontaine neighbors reading the same article can engage in a discussion and exchange opinions. News can nurture a sense of community. However, communities across the world with different cultural backgrounds and life-styles rarely feel connected. Because natural catastrophes occur on every continent with surprise and destruction, they offer an opportunity to point to similarities between communities. PLUM creates *connectedness* by changing the perspective of disaster articles.

Personalizing Information

Information can be tailored to a desired 'look', or style. Weitzman and Wittenburg [Weitzman 1995] present a system that generates different spatial layouts and graphical styles for the same multi-media document. Using a visual grammar based on one document's style, another document can be laid out in the same style. For example, their system transforms the table of contents of the conservative Scientific American to look like one out of WIRED, a publication with an avant-garde lay-out. While the content remains the same, the style of the presentation is tailored for a specific purpose or reader.

Information can also be tailored to fit a reader's expertise. Expert systems, computer programs that answer questions about a narrow subject, adapt to the user's level of knowledge. A tailored presentation by an expert system should not present any information obvious to the user or include facts the user cannot understand. For example, TAILOR [Paris 1993] describes an engine in terms appropriate for a hobbyist or an engineer.

In the past, cost prohibited tailoring news. Traditional print media sees its audience as a mass sending the same printed message to all readers [McQuail 1987]. Some newspapers manage to publish a regional issue. The New Jersey Journal recently launched a newsletter for the Indian community. The India Journal includes international news articles related to India and combines them with locally written ones. Digital media makes tailored news possible for all communities at insignificant expense. A common tailoring technique for digitally distributed news matches articles to a reader's interest model [Yan 1995]. This filtering can save the reader's time. However, filtered news may sacrifice the diversity of information if all articles outside the predefined focus are rejected.

PLUM does not need to accept or reject news. Concentrating on just one subject, PLUM personalizes news through augmentation. It does not maintain personal

records and thereby risk readers' privacy. PLUM operates on the reasonable assumption that residents know a common set of facts about their community.

Making Text More Meaningful

To make articles more meaningful to a reader, PLUM borrows techniques outlined in Richard Wurman's book, Information Anxiety [Wurman 1989]. Wurman points to the need for a 'personal media-measuring stick.' He wants to turn 'statistics into meaningful numbers, data into information, facts into stories with value.' Wurman seeks common sense explanations. As an example of a meaningless entity, Wurman takes the size of an acre. If he explains that an acre equals 43,560 square feet, we still cannot imagine its size. But, if he says it is roughly the size of a football field, we have a good idea. While Dora understands an acre as a football field, Heikki in Finland understands it as a soccer field. While PLUM tries to make culturally sensitive comparisons, it has no common sense. To explain that flood waters in Vietnam rose high enough to cover Boston's Longfellow Bridge, PLUM would have to know what a bridge is. Though CYC [Lenat 1990], a common sense data base, is under construction, it is not available. PLUM uses existing data bases of geographic and demographic facts to draw comparisons to facts residents probably know.

PLUM makes a number of comparisons to make readers understand a disaster. It finds a similar disaster that has occurred in their community or nearby. It overlays a shadow showing the extent of destruction on the home town map. PLUM compares the number of affected people to the home town population. PLUM calculates how much all home community residents would have to pay to cover the damage and what percentage of the city budget this represents. It also notes if any people from the affected country live in the community.

Arguably, no 'right' way exists to compare facts from different cultures and societies. Facts are sensitive to context and subject to interpretation. For example, if an article reports the evacuation of 500 families in Vietnam, a comparison cannot be readily drawn in Boston. In Vietnam, families may include several generations of relatives, but many American families are nuclear or separated. PLUM cannot resolve this problem. Nor can it represent the value of money in different cultures. Simply converting between currencies does not suffice. A farmer in a subsistence economy who loses a buffalo and a plough has lost his livelihood. In dollars, this may amount to $200. The equivalent in Bellefontaine would mean the loss of Dora's hardware store. Since PLUM cannot reason this way, it compensates by proposing several different interpretations for the facts in the article. This is illustrated in the Augmented News in Fig. 1.

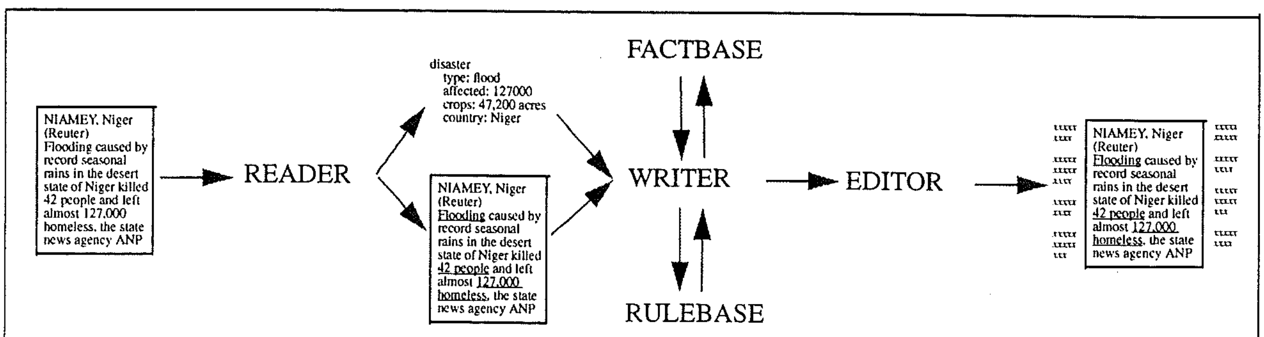
Description of the PLUM System

PLUM consists of five components. The FactBase contains all descriptive and statistical resources. The Reader parses in-coming disaster news. The RuleBase contains the rules for augmenting and the Writer executes these rules to produce the augmented text. The Editor lets a journalist make final modifications and generates the article Dora sees on her computer. The components of PLUM are illustrated in Fig 2.

Reader

Text processing by computer is difficult. A computer program cannot infer the meaning of natural language accurately. Language in free text is ambiguous and sensitive to context. However, stylized text, with a restricted vocabulary or sentences that follow patterns, allows a more accurate extraction of content. On-going research in the Machine Understanding Group at the Media Lab

Fig. 2



addresses text comprehension by computer. Haase's multi-scale parser [Haase 1993] uses word class and relations between words to determine their possible roles in a sentence. The Reader employs the multi-scale parser.

FRUMP [DeJong 1979], an early natural language understanding system, skimmed and summarized disaster news. FRUMP employed sketchy scripts. These sketchy scripts described the most likely events reported in a disaster article. Similarly, the goal of the Reader is not to understand every detail in an article. It tries to extract features such as the type of the disaster, the region or country it struck, the measured strength or extent, the damage to people, material, buildings, livestock, and the estimated costs. The Reader stores the extracted information in a template, as shown in Fig. 3.

Wire services use a relatively narrow vocabulary to describe disasters. This allows the Reader to use simple keyword techniques to extract some of the features. By combining keyword extraction with multi-scale parsing, the Reader achieves a higher rate of success. The example sentence below illustrates how the Reader processes text.

`Flooding caused by record seasonal rains in the desert state of Niger killed 42 people and left almost 127,000 homeless, the state news agency ANP said Friday.`

The Reader detects the type of the disaster -- flooding -- as a keyword and as equivalent to 'flood'. It detects the location of the disaster -- Niger -- as the name of a country occurring early in the article. It extracts the number of people killed -- 42 -- as part of the noun phrase following an active verb, possibly the object of the verb, 'kill'. '42' also modifies 'people'. The Reader finds the number of people affected by the disaster -- 127000 -- because it modifies 'homeless', a keyword describing people affected by a disaster.

Naturally, disaster articles report more than PLUM's template detects. An article may report on local and foreign aid issued, reasons the disaster was extraordinary, the history of disasters in the area, or quote a survivor. Difficulties arise if features are worded unpredictably or do not involve numbers or proper names.

FactBase

The FactBase contains information on the readers' home communities, the disaster-struck countries and the history of natural disasters. Two large cities, Boston, Massachusetts, and Helsinki, Finland, and a small rural town, Bellefontaine, Ohio, serve as example home communities. Using US and Finnish census data, the FactBase breaks the numbers down for each city's population by origin, language, occupation, income level. The FactBase includes data, maps, and distances between points at the city, county, state, and country level. The CIA World Fact Book provides background information on all countries. As illustrated in the Augmented News in Fig. 1, PLUM employs this data as a 'yard-stick' to help explain the impact of a disaster on a stricken country. PLUM also incorporates the CRED Disaster Database from the University of Louvain, Brussels, a history of disasters world-wide.

The FactBase is stored in FramerD [Haase 1995], a frame-based representation language. FramerD uses the Dtype library to construct data objects, ship them over networks, and store them in data base files. For example, each home community is an object in FramerD. The city, county, state, and country data are stored in separate frames inter-connected by relations 'part-of' and 'has-part'.

The lack of a standard data base format complicates the compilation of the data sources into one. While computers can access and let users read remote documents in the World Wide Web format, no standard exists to let computers access *and* read remote data bases. Data col-

Fig. 3

NIAMEY, Niger (Reuter) Flooding caused by record seasonal rains in the desert state of Niger killed 42 people and left almost 127,000 homeless, the state news agency ANP said Friday. Heavy rain since July has destroyed nearly 74,100 acres of crops and killed 6,800 animals, the agency said. The worst affected areas are the central Maradi region and the western Dosso and Niamey regions. The losses are estimated to be $52 million.

The average annual rainfall of 14 inches has fallen in a single day in some areas. On August 12, national television was forced to halt broadcasts because its studios were flooded.

DISASTER TEMPLATE

type: *flood*
location: *Niger*
killed: *42*
affected: *127000*
crops affected: *74100 acres*
livestock killed: *6800*
losses: *52000000 US$*

lections on the Internet contain little information about the structure of their content, often lacking even a standard delimiter between data fields or records. When making data sources available on the Internet, companies and academic departments need to accompany them with a description of the structure. Presently, every computer project tapping into an on-line data base probably wastes time reformatting the data for their own purpose.

RuleBase

The RuleBase defines how PLUM augments text. Each feature in the disaster template holds an augmentation rule. The rules describe how to compare distances, areas, quantities, and currencies in the disaster site against those in the home community. Other rules describe how to add background facts about the disaster-stricken country. The rules are not specific to a home community. PLUM can accommodate new communities and the rules still apply.

When PLUM fires, or executes, a rule, it uses the extracted disaster feature to search for comparable information in the FactBase. For example, in the news augmented for Bellefontaine, the following rule was fired to augment the number of people affected:

'Search through the home community, areas within it and areas to which it belongs. Look for a group of people in the same order of magnitude or a multiple of the number of people affected. Find the factor that makes the numbers equal.'

It produced: *'127,000 people is roughly three times the number of residents of Logan county.'*

The rules in PLUM have evolved during the design of the system, reflecting the feedback from students and faculty at the Media Lab. Since no straightforward recipe for augmenting exists, the rules represent one approach to localizing news.

Writer

The Writer fires the augmenting rules and adds the generated augmentations as hyper-links to the original article. For effective prose, language generation systems have to produce rhetorical structure for the text [Dale 1990]. PLUM avoids this because it does not rewrite the original story. Instead, it annotates the story by filling in sentence templates with appropriate information.

Augmentations alter the reading of the disaster story. The original linear story becomes a hyper-text document. Dora can explore the augmentations of her choice or view all the augmentations at once. Since the Writer makes three augmentations of the same article, one for each community, Dora can also read about the Niger flood from Heikki's perspective in Helsinki.

The Writer also generates a map of the reader's home town overlaying a representation of the destruction caused by the disaster overseas.

Editor

PLUM has no common sense. However accurate its parsing or augmenting rules may be, unpredictable turns of phrases can result in erroneous parsing and nonsensical augmentations. PLUM was designed as a computer tool for a journalist. Using PLUM's Editor interface a journalist, can accept, reject or modify the proposed augmentations. He has the final say in the augmented article. The Editor also allows the journalist to browse the background resources, do on-line research, add to the FactBase and compare raw statistics between cities.

How Robust is PLUM?

PLUM, a project in progress, will be evaluated at two levels, by measuring its robustness as a system and its usefulness as judged by readers.

If the Reader is robust, it will accurately extract the disaster features from articles not used as models in the design. The parsing algorithms were designed using 50 sample articles. They will be tested on 50 random disaster articles. Similarly, if the Writer, the RuleBase and the FactBase are robust, they will accurately augment the new articles.

Hopefully PLUM will provide readers new perspectives on disaster news. No obvious technique exists to quantify its success. The only feedback will be readers' comments. They will judge if augmented articles are more relevant and informative. They can also remark which kinds of augmentations are most useful.

Augmenting Beyond Disasters?

Presently, PLUM only augments disaster articles. Such news lends itself to this kind of annotation. Its stylized reporting allows a fairly accurate automatic processing. Several on and off-line resources for background information and statistics are also available. Since disaster news often leaves positive facts unreported and evokes misconceptions, it should be improved. PLUM cannot process less stylized news topics accurately. If natural language processing techniques improve or aided by a journalist, it could process other domains.

PLUM only augments for geographic communities.

It could be programmed to augment for other types of communities. For example, fishermen all share a knowledge of the weather, the sea, and migrations of fish. If PLUM contained this common knowledge, it could make disaster news relevant to fishermen, and with better parsing, news on other topics.

Conclusion

PLUM belongs to a new generation of computer tools for journalists and readers. It helps a journalist improve the content of his newspaper. PLUM automatically localizes news wires on disasters and using community profiles, makes the news more accessible to residents. By pointing to similarities between one's home town and an otherwise remote disaster, it creates a sense of connectedness.

References

[Cable & Broadcasting 1994] Cable & Broadcasting, Issue of October 31, 1994.

[Cate 1993] Fred H. Cate. Media, Disaster Relief and Images of the Developing World. Publication of the Annenberg Washington Program, Washington D.C., 1993.

[Dale 1990] Robert Dale, Chris Mellish, Michael Zock (eds). Current Research in Natural Language Generation. Academic Press, 1990.

[DeJong 1979] Gerald DeJong. Script application: Computer understanding of newspaper stories. Doctoral Thesis, Yale University, New Haven, 1979.

[Haase 1993] Ken Haase. Multi-Scale Parsing Using Optimizing Finite State Machines. ACL-93 Proceedings, 1993.

[Haase 1995] Ken Haase and Sara Elo. FramerD, The Dtype Frame System. MIT Media Lab internal report, 1995.

[Lenat 1990] D.B. Lenat and R.V. Guha. Building Large Knowledge Based Systems. Addison-Wesley, Reading, MA, 1990.

[McQuail 1987] Denis McQuail. Mass Communication Theory. Sage Publications, 1987.

[Schank 1990] Roger Schank. Tell Me a Story. Charles Scribner's Sons, 1990.

[Weitzman 1994] Louie Weitzman and Kent Wittenburg. Automatic Representation of Multimedia Documents Using Relational Grammars. ACM Multimedia'94, San Francisco, 1994.

[Wurman 1989] Richard S. Wurman. Information Anxiety. Doubleday, 1989.

[Yan 1995] Tak Woon Yan and Hector Garcia-Molina. SIFT -- A Tool for Wide-Area Information Dissemination. Proceedings of the 1195 USENIX Technical Conference, pp. 177-186, 1995.

Toward a Taxonomy of Logical Document Structures

Kristen Summers
Department of Computer Science
Cornell University
Ithaca, NY 14853
summers@cs.cornell.edu

Abstract

The automated discovery of logical structure in text documents is an important problem that has recently received a good deal of attention; it can enable the creation of flexible and sophisticated document manipulation tools that will greatly increase the impact of electronic documents. This paper addresses aspects of the nature of the logical structures to be found, in order to develop categories of structures that reflect the variance in requirements for discovery and the variance in significance for applications. A complete taxonomy is not developed, but relevant attributes are identified in three forms of categorization: fundamental, based on structure definitions; discovery, based on required observables to find structures; and usage, based on roles structures play in applications. The attributes themselves are independent of the choice of particular logical structures to consider in a given application, and their direct implications are discussed.

1 Introduction

The automated discovery of logical structure in text documents is an important problem that has recently received a good deal of attention. A solution to this problem would, based on a representation of the physical instantiation of a document, create a hierarchy of the logical components of the document: paragraphs, sections, lists, etc. This hierarchy can enable a variety of applications in the realm of information access, including browsing, retrieval, and automated hyperlinking.[1] These applications can provide flexible and sophisticated document manipulation tools that will greatly increase the impact of electronic documents. This paper describes categorizations of logical components that will be useful both in designing solutions to the problem and in evaluating their performance.

1.1 Logical Document Structure

A logical structure tree for the present paper is given in Figure 1. (Other trees may be formed by including different degrees of granularity or organizing the components differently.) It should be clear that browsing may proceed based on tree navigation [4], hyperlinking may be performed by observing significant relationships between node values [1], a form of retrieval may be achieved by specifying tree locations of interest (and attributes they must have) [5, 13], the reuse of document portions will be eased by this kind of retrieval [15], and multiple style instantiations of the same document can be achieved by applying the corresponding style rules to a single tree [2, 8].

This logical structure should be distinguished from the *layout* structure that describes the physical text on the page and *content* structure that describes purely semantic relationships within documents, as follows.[2]

Definition 1 *The* logical structure *of a document consists of a hierarchy of segments of the document, each of which corresponds to a visually distinguished semantic component of the document.*

That is, logical structure lies at the intersection of content and layout; a logical segment must both be distinguished by its layout (thus the concept of a cohesive text passage is content only) and have meaning as a semantic unit (thus boldfaced text is layout

[1] When a marked-up version of the document is available (in SGML or another format), with a known set of markers, then the logical hierarchy is directly available. The problem of discovery arises when this is not the case, e.g., when a document is scanned-in, or when the available representation is in a Page Definition Language like PostScript.

[2] [12] recommends further distinguishing the *geometry* of a document, which includes line breaks, page breaks, etc., from the *layout*, which describes only the formatting guidelines, such as left-justification.

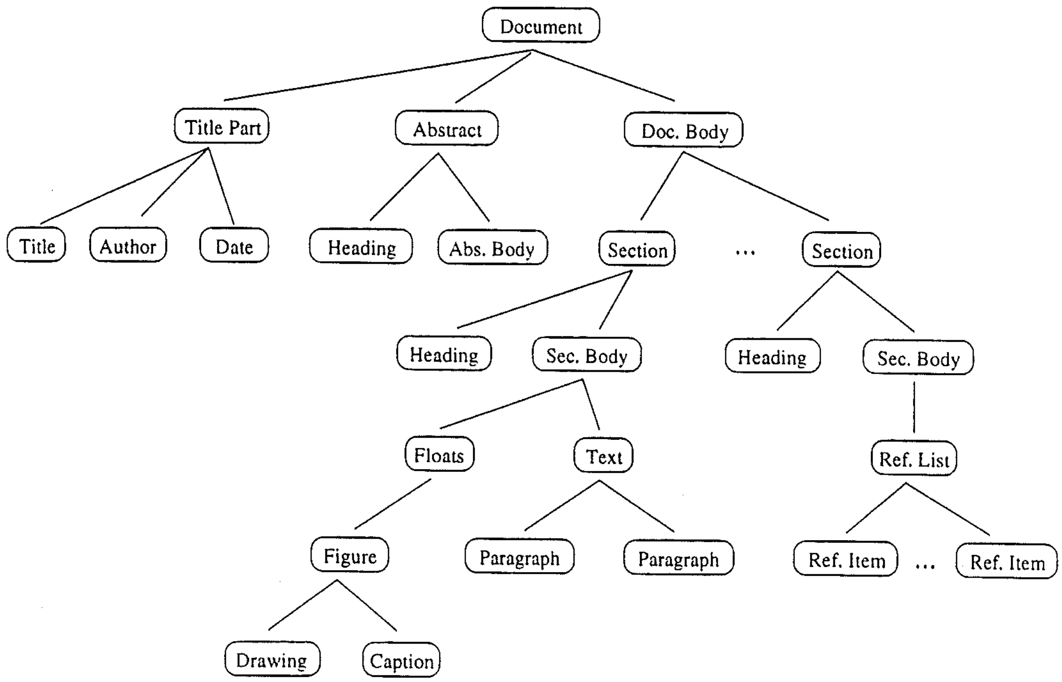


Figure 1: A logical structure tree for the present paper

only). Furthermore, the logical structure of a document refers to the hierarchy formed by the containment relationship among these components; other relationships exist, such as that formed by references in the text, but these do not form a part of the logical structure in the current sense.[3] Each type of element in such a hierarchy is also referred to as a logical structure.

1.2 Typing Logical Structures

Work in the area of logical structure discovery and use raises interesting questions about the nature of these structures. Can we find classes of structures that correspond to degrees of difficulty in their discovery, based on the necessary observables and/or a space-time complexity kind of analysis? Can we find classes of structures that correspond to their significance for the user? This paper does not provide a complete answer to these questions, but it does identify many attributes that will need to be considered in the attempt to find such an organization of logical document structures. These attributes and categorizations exist independently of the choice of particular structures to consider in a given application. The categorization of a particular logical structure does depend on its definition, which may vary in different contexts and applications; the examples in this paper are intended to follow sufficiently general rules as to be applicable in most settings.[4]

The following subsection discusses related work; the remaining sections discuss different forms of categorizations of logical document structures. These logical structures can be categorized in at least the following ways: fundamental distinctions, based on the definitions of the structures themselves and discussed in Section 2; discovery distinctions, based on the observable characteristics required (or helpful) for identifying the structures and discussed in Section 3; and usage distinctions, based on the kinds of use of the structures that can be expected in applications and discussed in Section 4.

[3] The approaches to logical structure discovery cited here focus on the problem of finding the logical structure in this sense (called the *primary structure* in [10]); other relationships can often be derived directly from a combination of this tree, effective optical character recognition, and a small amount of text analysis.

[4] For instance, an assumption in this paper is that the structures theorem and definition should be distinguished but are not formatted differently according to any predefined rule; there may be contexts in which they are formatted differently and this is known *a priori*, but this is not typical.

1.3 Related Work

Much work in the area of logical document structure focuses either on its automatic discovery or on application of the information contained in this structure, and some work addresses the question of how these structures should be represented.

Discovery Approaches to logical structure discovery assume some prior knowledge of the style of the document, i.e., the effects of logical structure on layout; layout observations are then analyzed to determine the logical structures that caused them. The required information about the document style ranges from very specific and precise to fairly general ideas about the ways in which it is possible to convey logical information through formatting. This information is presented as a grammar, and the document layout is parsed in [3, 9, 11, 12]. Other approaches of varying degrees of similarity to parsing and based on varying degrees of knowledge specificity, are presented in [6, 7, 16, 18, 21, 22, 23, 24, 25].

Applications Applications of the solution (which may be represented as a separate hierarchy, with pointers to document locations, or as a marked-up version of the document) are discussed in [1, 2, 4, 5, 13, 15, 19, 20, 26]. These applications include, but are not limited to, those discussed in Subsection 1.1.

Representation Discussions of types of logical structure representation can be found in [10, 17]. [10] provides a taxonomy of full structural hierarchies, considering attributes such as the choice of atomic unit of structure. [17] formalizes and extends ODA (Office Document Architecture) in an object-oriented framework; in the process, an object taxonomy is presented that distinguishes between layout and logical objects and between simple and composite objects. Further distinctions among logical object classes are not explored, as they do not affect the goal of the paper.

This paper differs from the above work in that its concern is with the nature of the units of logical structure themselves. Since the attributes of logical structures have implications for the kind of work described above, it is to be hoped that the current interest in finding and using logical structures will lead to more complete explorations of the nature of these structures.

2 Fundamental Divisions

The most basic divisions of logical structures rest on the definitions of the structures themselves. These distinctions have obvious, direct implications for the structure discovery; what is included in the definition of a structure affects the preferred method of identifying this structure.[5] They also affect the other categorizations. These divisions include distinctions between primary and secondary structures and between content-oriented and layout-oriented structures.

2.1 Primary vs. Secondary

Primary structures are defined, at least in part by their own attributes; *secondary* structures can be completely defined by their positions in the hierarchy and relationships to other structures. For example, a section heading is a primary structure; it is identifiable by its appearance and separation from the surrounding text. This primary structure provides the basis for finding the secondary structures section body and section. A section body is a right sibling of a section heading with, in turn, no right sibling of its own;[6] a section is a node whose children are exactly a section heading and a section body. Figure 2 shows Figure 1, with primary structures in solid boxes and secondary structures in dashed boxes.

2.2 Content- vs. Layout-Orientation

Another fundamental distinction can be made based on the relative roles of content and layout in the definition of a logical structure. Although both must be included, some logical structures can be considered *content-oriented*, and some can be considered *layout-oriented*. For example, a definition is a logical structure when it is distinguished by its presentation, as in Section 1 of this paper; it remains, however, a content-oriented structure. On the other hand, a special paragraph (a paragraph presented in other than the usual format for a given document[7]) is a layout-oriented structure. These descriptions are relative; a logical structure is more content-oriented than another if its definition relies more heavily on internal

[5] The structure definition does not completely determine how to discover it, however; extra-definitional cues may be quite useful, and at times it may be appropriate to categorize a document piece as a structure whose definition it does not match precisely.

[6] This definition refers to an ideal tree, in which the sections have been correctly identified. In the process of forming a tree with an imperfect method, a more useful definition might be: a section body is a right sibling of a section heading whose own right sibling, if it exists, is also a section heading.

[7] In this paper, definition is a subtype of special paragraph.

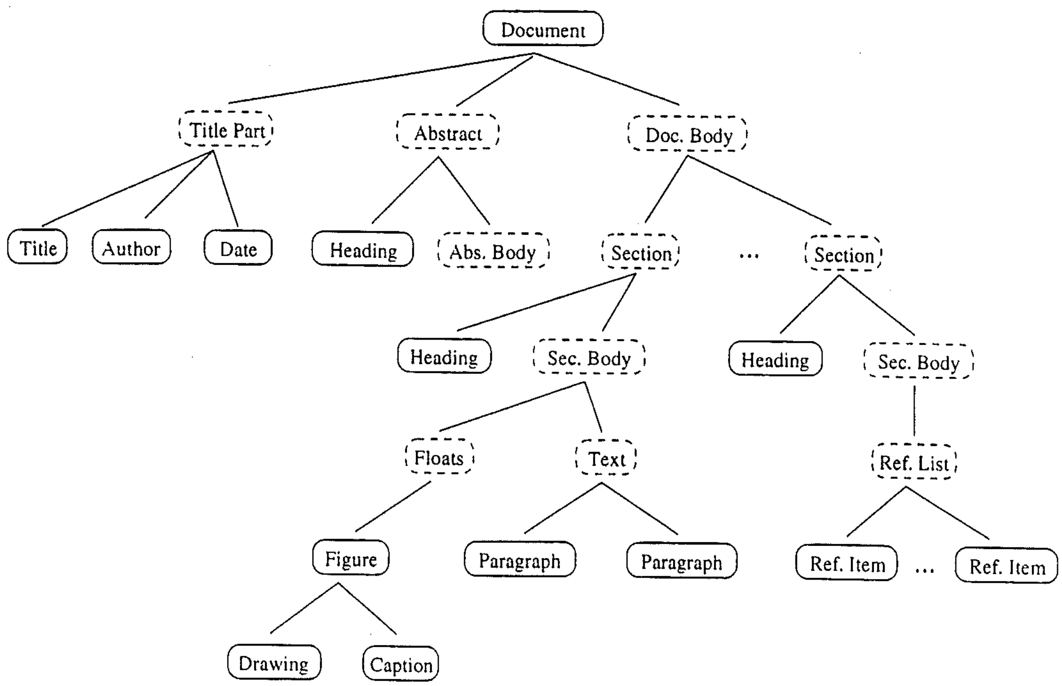


Figure 2: The earlier tree, with primary and secondary structures distinguished

meaning; similarly, a more layout-oriented structure has a definition that relies more heavily on visual presentation.

To make this precise, consider the hierarchy that can be formed among logical structures themselves, in which the children of a node are subtypes of the node's structure.[8] A portion of this hierarchy is given in Figure 3. If a structure is distinguished from its siblings entirely by content, it is content-oriented; if it is distinguished from its siblings entirely by layout, it is layout-oriented; otherwise, it is neither.

Degrees of this kind of orientation are distinguished by the degree to which this definition can be extended. That is, a structure that is distinguished from its siblings and its parent's siblings by content alone is more content-oriented than one that is distinguished from its own siblings by content alone but from one or more of its parent's siblings in part by layout. (Note that if a node is distinguished from its parent's siblings by content alone, it is therefore also distinguished from its first cousins, i.e., its parent's siblings' children, by content alone.) A structure that is also distinguished from its grandparent's siblings by content alone is more content-oriented still, etc. This is equivalent to the idea that the degree of content-orientation corresponds to the number of immediate ancestors of a content-oriented structure that are also content-oriented. Of course, this definition of degrees of orientation also applies analogously to layout-orientation.

[8] Typically, the logical structure of a document will be given in terms of the structures at the leaves of this hierarchy.

3 Discovery Divisions

Primary logical structures can be characterized by the cues that are necessary and/or useful in their discovery. (Secondary structures can, of course, be discovered by applying their definitions after primary structures have been found; thus, no additional cues are needed.) These cues belong to four basic categories: *geometric*, *marking*, *linguistic*, and *contextual*.

Table 1 at the end of this section provides several examples of necessary and useful discovery cues for primary logical structures, in terms of these categories. Necessary cues are marked as "Nec.," and useful cues are marked as "Helps." Note that a structure must require at least the observables required by its ancestors in the structure hierarchy of Figure 3.

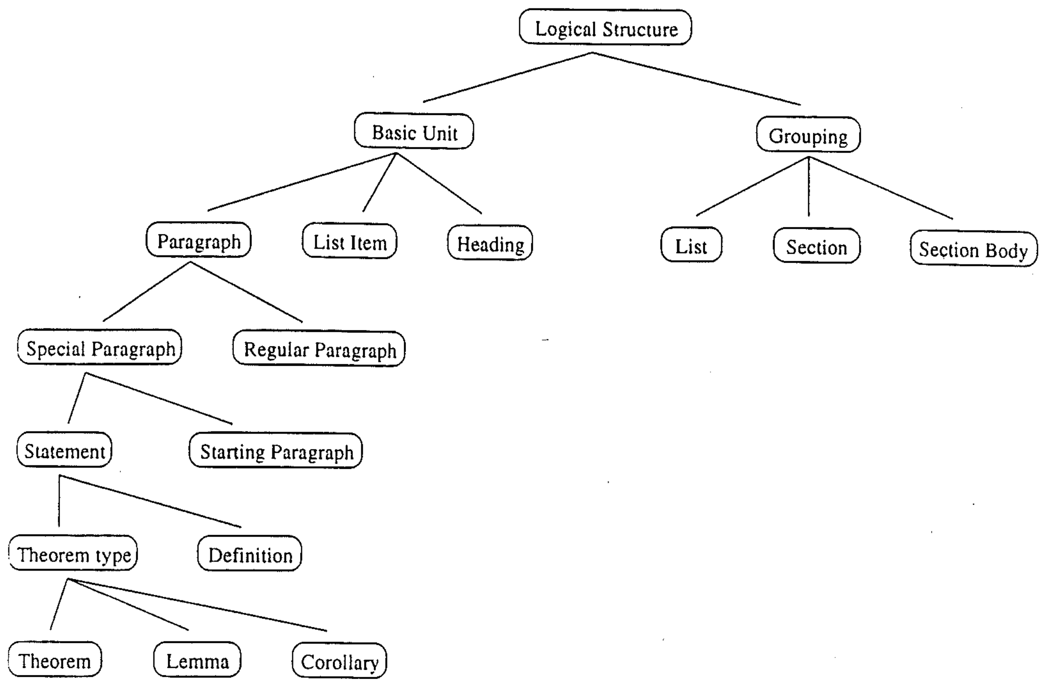


Figure 3: A partial hierarchy of logical structures

3.1 Geometric Observables

Geometric observables include the (external) *contours* and the *internal shape* of a piece of text. (*Height* is a special case of *contours*.) Both of these kinds of cues may be necessary; for instance, the *contours* of an indented list provide its characteristic shape of a hanging indent, but the shape of a table is recognized by the *internal shape* of its columnization [14]. Since geometry involves the shapes formed by the marks on the paper or screen, its contribution can (inversely) be found by an analysis of the white space in a document [21].

3.2 Marking Observables

Marking observables consist of non-linguistic marks on the paper or screen; this includes attributes like font type and weight, as well as non-alphanumeric *symbols*, such as bullet points and rule lines. Bullet points and dashes, for instance, can aid in the identification of indented list items; symbols can be necessary to find left-justified list items.

For example, consider Figure 4, in which a portion of an actual e-mail message is represented with different sets of observables. In all cases, alphabetic characters have been replaced with the letter "x." In the upper left version, all symbols and characters have been so replaced; in this representation, no difference in the format of the text blocks is visible. In the upper right version, the observables include *symbols*; the lower two text blocks can be observed to begin with a parenthesized character, suggesting that they are items in a left-justified list. In fact, this is so, as is quite clear in the lower right version, in which both *symbols* and numbers are included. (In this case, either *symbols* or numbers are sufficient to suggest the presence of a list without the other, but either may be required, depending on the form of marking the list items.)

3.3 Linguistic Observables

Linguistic observables include combinations of *numeric* and *alphabetic* symbols. (These cues enter a gray area between *symbolic* and *linguistic* when they are character-based rather than word-based.) The observation of words is necessary for structures such as theorem to be recognized and distinguished from similar structures (e.g., definitions, in many cases). The identification of indented list items is aided by *numeric* cues just as by *symbolic* ones; again, these cues can be necessary to identify justified list items. For example, if the item numbers in Figure 4 were

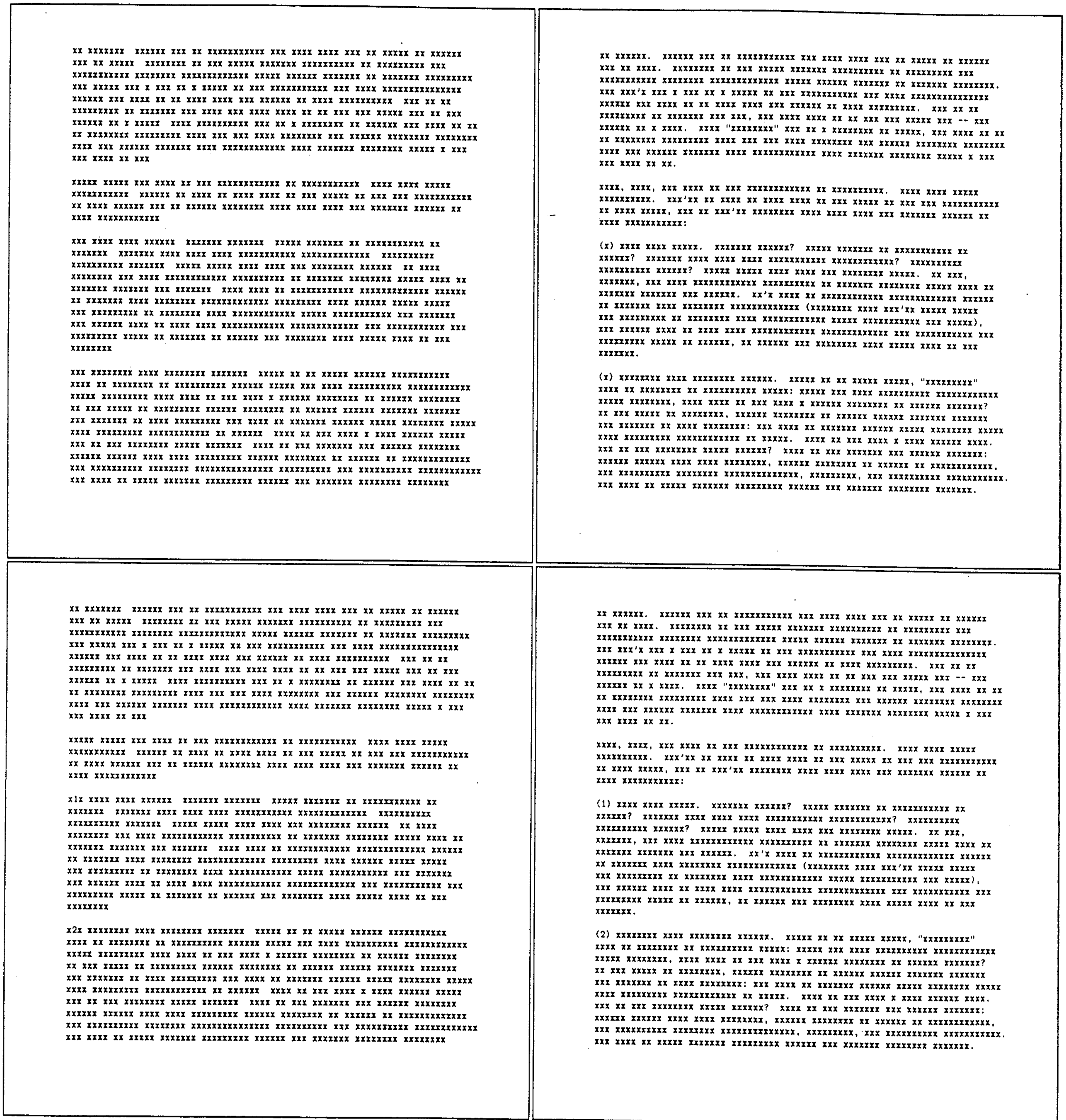


Figure 4: Paragraphs and a justified list, with and without observable symbols and numbers

not enclosed in parentheses, *symbols* would not identify the list, and *numeric* cues would be necessary to find it. In its current form, *numeric* cues are enough to suggest that it is a list, as can be observed from the lower left representation.

Typically, content-oriented structures will require linguistic cues, since content is usually contained in the language of a document. This linguistic analysis can remain quite shallow or become very complex; naturally, the subtlety of the content aspects of discoverable structures depends in part on the depth of the analysis. For example, consider an attempt to distinguish an author's institutional affiliation (one structure) from address (another structure), without making use of further analysis than checking for the presence of keywords. In order to find most institutions, the relevant keywords would probably include "University, College, Corporation, Company," to name but a few; the effect would be that streets like "University Avenue" and towns like "College Park" would be incorrectly identified as affiliations.[9] The goal is probably not reasonable for the amount of included analysis.

3.4 Contextual Observables

Contextual observables can be divided into *local* and *global* context-based cues. Local contexts use information about some limited number of surrounding nodes: siblings, parents, children, or neighbors within a level (which may or may not be siblings). For instance, consider a typical business letter; the return address and the closing (including the signature, etc.) are both internally left-justified blocks indented approximately halfway across the page; in this setting, they can be distinguished easily by local context, since the return address is not preceded by any text, but the closing is.

Global contexts use information about the document as a whole. For example, a special paragraph is a paragraph that differs in its presentation from the typical paragraphs within the document.[10]

Contextual information may, of course, include information of any of the preceding varieties; moreover, it may make use of available structure type information.

[9] Some cases could be filtered out by requiring that affiliations not contain numeric values, but this would not cover every case.

[10] Identifying the typical is a significant problem, as the standard may not always occur more frequently than the nonstandard.

4 Usage Divisions

The logical structures of a document may also be characterized according to their use. This kind of categorization attempts to capture information about the relative significance of different logical structures; it has implications for performance evaluation of logical structure discovery.

The relative importance of logical structures is, of course, application-specific. As an extreme example, consider the application of a theorem extractor. For this tool, theorem is the only structure of direct significance. Its ancestors in the structure hierarchy are also important, to the extent that errors in their identification may lead to errors in identifying theorems; no other structures matter, so errors in their identification are insignificant. Finding the full logical structure would be overkill for this application, but the point stands that if a structure discovery mechanism is designed for a particular application, its output should be evaluated with respect to structure importance within that application.

In the more general case, however, the logical structure is derived for possible use with many applications, and the kind of information described above is therefore unavailable. A more general (and necessarily less precise) concept of structure significance is required. This raises a variety of different issues, including: classifier implications, expected user references, hierarchy role, and generality. These are described below.

Table 2 at the end of this section provides several examples of the usage characteristics of logical structures, in terms of these categories.

4.1 Classifier Implications

This attribute refers to the relevance of a structure to the identification of other structures. To some extent, this is dependent upon the classifier itself, but it also depends on the intrinsic definitions of the structures; the definitions of secondary structures highlight the importance of certain other structures, as do the definitions of primary structures that depend in part on their contexts (e.g., special paragraph relies on other paragraphs). For example, section headings are quite significant in this respect, as two secondary structures (section bodies and sections) rely on them for correct identification.[11]

[11] Here, as elsewhere in this paper, the word "section" may be replaced by "subsection" or "sub^{n}section" where $n \geq 0$.

Table 1: Some Primary Structures and Discovery Cues

Structure	Geometry		Marking		Linguistic		Context	
	Contour	Internal	Font	Symbol	Word	Number	Global	Local
Paragraph	Nec.	Helps		Helps				
Special Paragraph	Nec.	Helps	Helps	Helps			Nec.	Helps
Theorem	Nec.	Helps	Helps	Helps	Nec.		Nec.	Helps
Indented List Item	Nec.		Helps	Helps		Helps		Helps
Justified List Item	Nec.		Helps	Nec.		Nec.		Helps
Indented List	Helps							Nec.

4.2 Expected User References

Structures that users refer to more frequently than others are, in an important sense, particularly significant. For example, if users write queries that ask for full sections more often than groups of paragraphs, then sections are more significant for retrieval than are paragraphs (unless, of course, these sections are frequently identified by particular paragraphs they contain).

This attribute is task-dependent; different structures may be commonly used in retrieval from those commonly used in browsing, for example.

Determining precise expectations for this would require tracking the behavior of actual users with a fully general system (including different kinds of documents). Such a study would provide a solid basis for according relative weights to different logical structures, with respect to this attribute.

In the absence of strong empirical evidence, however, certain general observations can be made, based on the natures of commonly suggested applications.

- In hierarchical browsing as described in [4, 21], structures at higher levels of the tree are more significant than those at lower levels. Since this browsing is based on tree navigation, starting from the root, higher-level structures will be used for earlier decisions, on which later decisions will in part rely. Furthermore, for any node that is accessed, all of its ancestors must have been accessed as well, but its descendents need not be. So for this application significance corresponds (to a large degree) to height.

- For hyperlinking, bibliographic structures have a special significance. Links will often be desirable based on bibliographic matches (such as articles that share authors, or a match between a reference in one article and the title of another), so the structures that provide this information are particularly important. Floats (figures, tables, etc.) have a similar importance for linking, as they are typically referenced in the text.

  More conceptual hyperlinks are often desirable as well, of course. It is not obvious how these will relate to logical structure, however, as the criteria for their inclusion are still emerging.

- In the retrieval of previously-seen documents or document portions, highly salient structures are likely to be those which differ greatly from their surroundings, thereby calling attention to themselves. (This is an intuition, which can be verified or disproved by tracking use of a system with logical structure-based retrieval. Furthermore, determining which structures differ greatly from their surroundings in this sense is far from trivial.)

The above is not meant to be exhaustive; it simply provides an example of the kinds of issues that can provide insight into the significance of different logical structures from the direct perspective of a user.

4.3 Hierarchy Role

A significant distinction can be drawn between those structures that exist in order to express a useful piece of the document and those that exist in order to complete the hierarchy. For example, the structure paragraph part is not, in itself, useful; it exists in order to complete the children of a paragraph that contains an equation or an indented list or some other interruption. *Filler* structures that exist only to complete the hierarchy are a proper subset of secondary structures. Useful structures are, of course, more significant than are fillers (although distinguishing the two may be extremely significant!)

Table 2: Some Structures and Usage Characteristics

Structure	Implications	Task Importance	Hierarchy Role	Generality
Paragraph	Para. Group		Useful	High
Heading	Section Body, Section	Browsing	Useful	High
Section		Browsing	Useful	High
Section Body		Browsing	Filler	High
Title Part	Title, Author, etc.	Browsing	Useful	High
Title		Browsing, Linking	Useful	High
Corollary	Proof		Useful	Low

4.4 Generality

Consider the structure hierarchy, partially shown in Figure 3. Distinguishing between structures that appear at lower levels of the tree changes the meaning of the result less than distinguishing between structures at higher levels; thus those at higher levels are more significant, in that their correct identification provides more new content.

5 Conclusions

This paper has described several criteria for categorizing the types of logical structures of text documents. Although a general system for typing these structures has not been achieved, many issues have been raised that must be considered in the process. These fall into three (sometimes interrelated) categories: fundamental distinctions, based on structure definitions; discovery distinctions, based on necessary and useful cues for structure identification; and usage distinctions, based on structure roles in applications. Identifying these attributes and differences provides an important step towards developing a more general theory of classes of logical document structures. This problem deserves further attention, as its solution will have significant implications for the development and evaluation of logical structure discovery techniques.

Acknowledgments

I am very grateful to John Hopcroft for proposing the problem and for his guidance and support; I am also very grateful to Daniela Rus for her guidance. Special thanks to Jim Davis for a careful reading of an early draft and to and all members of the Information Capture and Access group at Cornell for enlightening discussions. Thanks also to the anonymous reviewers for many helpful comments.

References

[1] James Allan, Jim Davis, Dean Krafft, Daniela Rus, and Devika Subramanian. Information agents for building hyperlinks, 1993.

[2] Dennis S. Arnon. Scrimshaw: A language for document queries and transformations. *Electronic Publishing*, 6(4):385–396, 1993.

[3] Henry S. Baird. Anatomy of a versatile page reader. *Proceedings of the IEEE*, 80(7):1059–1065, 1992.

[4] Victoria A. Burrill. VORTEXT: VictORias TEXT reading and authoring system. In J. C. van Vliet, editor, *Text Processing and Document Manipulation: Proceedings of the International Conference*, British Computer Society Workshop Series, pages 43–57, Nottingham, April 1986. Cambridge University Press.

[5] Charles L. A. Clarke, G. V. Cormack, and F. J. Burkowski. An algebra for structured text search and a framework for its implementation, August 1994. URL: `ftp://cs-archive.uwaterloo.ca/cs-archive/CS-94-30/structxt.dvi`.

[6] Denise Derrien and Michel Habib. Approche objet pour l'analyse de la structure logique des documents. In Jacques André and Jean Bézivin, editors, *Woodman '89: Workshop on Object-Oriented Document Manipulation*, pages 226–235, Rennes, May 1989.

[7] Floriana Esposito, Donato Malerba, and Giovanni Semeraro. Multistrategy learning for document recognition. *Applied Artificial Intelligence*, 8:33–84, 1994.

[8] An Feng and Toshiro Wakayama. SIMON: A grammar based transformation system for structured documents. *Electronic Publishing*, 6(4), 1993.

[9] Hiromichi Fujisawa, Yasuaki Nakano, and Kiyomichi Kurino. Segmentation methods for character recognition: From segmentation to document structure analysis. In *Proceedings of the IEEE*, volume 80, pages 1079–1092, July 1992.

[10] Richard Furuta. An object-based taxonomy for abstract structure in document models. *The Computer Journal*, 32(6):494–504, 1989.

[11] Tao Hu and Rolf Ingold. A mixed approach toward an efficient logical structure recognition from document images. *Electronic Publishing*, 6(4):457–468, 1993.

[12] Rolf Ingold. Text structure recognition in optical reading. In Jacques André, Richard Furuta, and Vincent Quint, editors, *Structured Documents*, The Cambridge Series on Electronic Publishing, pages 133–141. Cambridge University Press, Cambridge, 1989.

[13] Michael H. Kay. Textmaster – document filing and retrieval using ODA. In J. C. van Vliet, editor, *Text Processing and Document Manipulation: Proceedings of the International Conference*, British Computer Society Workshop Series, pages 125–139, Nottingham, April 1986. Cambridge University Press.

[14] Christopher Lewis, Daniela Rus, and Matthew Scott. A structure detector for tables. Forthcoming Technical Report.

[15] Keith McAlpine and Paul Golder. A new architecture for a collaborative authoring system: Collaborwriter. *Computer Supported Cooperative Work*, 2:159–174, 1994.

[16] Masaaki Mizuno, Yoshitake Tsuji, Toshiyuki Tanaka, Haruhiko Tanaka, Masao Iwashita, and Tsutomu Temma. Document recognition system with layout structure generator. *NEC Research and Development*, 32(2):430–437, July 1991.

[17] Makoto Murata. An object-oriented interpretation of ODA. In Jacques André and Jean Bézivin, editors, *Woodman '89: Workshop on Object-Oriented Document Manipulation*, pages 91–100, Rennes, May 1989.

[18] Gilbert B. Porter and Emil V. Rainero. Document reconstruction: A system for recovering document structure from layout. In *Proceedings of the Conference on Electronic Publishing*, pages 127–141, 1992.

[19] T. V. Raman. *Audio System for Technical Readings*. PhD thesis, Cornell University, May 1994. URL: `http://www.cs.cornell.edu/Info/People/raman/phd-thesis/`.

[20] Daniela Rus and Devika Subramanian. Multimedia RISSC Informatics: Retrieving Information with Simple Structural Components. In *Proceedings of the ACM Conference on Information and Knowledge Management*, Washington DC, November 1993.

[21] Daniela Rus and Kristen Summers. Using white space for automated document structuring. In *Proceedings of the Workshop on Principles of Document Processing*, Seeheim, 1994.

[22] Yasuo Tanosaki, Kenji Suzuki, Kiyoshi Kikuchi, and Motoshi Kurihara. A logical structure analysis system for documents. In *Proceedings of the Second International Syposium on Interoperable Information Systems*, pages 221–228, Tokyo, November 1988.

[23] Dacheng Wang and Sargur N. Srihari. Classification of newspaper image blocks using texture analysis. *Computer Vision, Graphics, and Image Processing*, 47:327–352, 1989.

[24] Toyohide Watanabe, Qin Luo, and Noboru Sugie. Structure recognition methods for various types of documents. *Machine Vision and Applications*, 6:163–176, 1993.

[25] K. Y. Wong, R.G. Casey, and F. M. Wahl. Document analysis system. *IBM Journal of Research and Development*, 26(6):647–656, November 1982.

[26] Haviland Wright. SGML frees information. *Byte*, 17, June 1992.

Two Digital Library Interfaces That Exploit Hierarchical Structure

Robert B. Allen
Bellcore
MRE 2A367
445 South Street
Morristown, NJ USA
rba@bellcore.com

1. ABSTRACT

Two library classification system interfaces have been implemented for navigating and searching large collections of document and book records. One interface allows the user to browse book records organized by the Dewey Decimal Classification hierarchy. A Book Shelf display reflects the facet position in the classification hierarchy during browsing, and it dynamically updates to reflect search hits and attribute selections. The other interface provides access to records describing computer science documents classified by the *ACM Computing Reviews* (CR) system. The CR classification system is a type of faceted classification in which documents can appear at several points in the hierarchy. These two interfaces demonstrate that classification structure can be effectively utilized for organizing digital libraries and, potentially, collections of Internet-wide information services.

2. CLASSIFICATION SYSTEMS FOR ORGANIZING LARGE ELECTRONIC INFORMATION ARCHIVES

2.1. Advantages of Classification-Based Interfaces

Organizing books and documents in a digital library interface by an a priori classification system may seem to be a weak alternative to the variety of ad hoc organizations possible in response to searches. However, a consistent structure, reflecting a commonly agreed upon organization of knowledge, may help orient the user. As suggested by Mann[15]:

> *Given identical computer systems for searching the catalog records, is there an additional and substantial advantage in being able to search the full texts themselves in subject-browseable groups?*
>
> *I submit that anyone who actually has to do research, especially in unfamiliar subject areas or in languages in which he* [sic] *has little proficiency, would have a decided and fully justified preference for working in Library A* [with subject-browseable groups]. (page 131).

Indeed, an interface which reflects the structure of the classification system essentially provides suggestions to a user about further options to pursue following a search. That is, after a search a user can select from the node labels of the classification system near the search hits to identify the subdivisions that may help further refine the search. The classification system can also be used to restrict searches so as to reduce the computational cost and avoid overwhelming users with spurious information.

This paper considers two types of interfaces for accessing books and documents organized in classification systems. The interfaces have been implemented in the X Window System using Motif widgets. The first interface (Section 2) is for the Dewey Decimal Classification (DDC). this uses the hierarchical organization to facilitate browsing and the presentation of book records. The second interface (Section 3) manages documents organized by a type of faceted classification system.

2.2. OPACs and Electronic Book Interfaces

Several interfaces have been developed for accessing online book records. However, most Online Public Access Catalog (OPAC) interfaces are designed for ASCII terminals and do not have advantages, such as direct manipulation, associated with GUIs. Other OPACs provide extensive term searching but do not take advantage of the hierarchical organization [8]. Book cataloging systems also provide access to the hierarchical classifications. However, these generally have only simple graphical interfaces (e.g., [18]) and are not documented in the literature. Some prototype electronic catalogs introduce creative interfaces but may not scale well for large collections [4, 17, 19].

Interfaces for electronic books have now been widely studied, but relatively little attention has been paid to the management of collections of books in these systems. The SuperBook^{TM} browser [6, 7] takes advantage of the hierarchical structure of individual documents. For instance, it presents chapter and section headings in a dynamic Table of Contents (TOC). However, the SuperBook browser itself is not effective for navigating a hierarchical book classification system; it does not easily support fielded search, and it is not designed for presenting and manipulating short records.

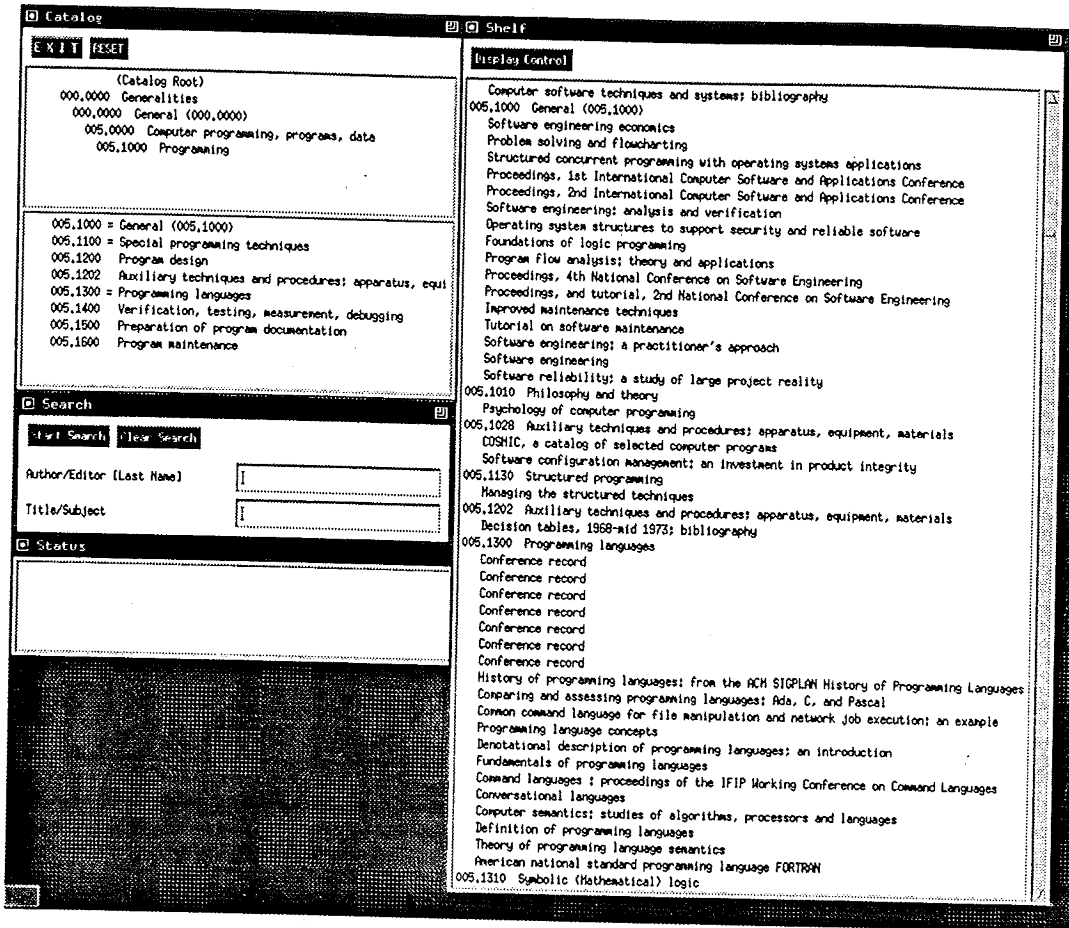


Figure 1: GUI for Book Records Organized by Dewey Decimal Classification.

Section 2 describes an interface that incorporates interface features from other systems and adds many new ones. Among these features are fisheye browsing of the classification hierarchy, a full Book Shelf, interlocking operation of the classification hierarchy and Book Shelf display, posting search hits against the classification hierarchy, control of search hit displays on the Shelf, control of the granularity of the search hit displays, and lateral links across the classification hierarchy. Moreover, it supports a realistically large collection of book records.

2.3. Interface for Faceted Classifications

Many classification hierarchies have multiple components. These include faceted classifications [21], polyhierarchies, and multitrees [10]. Faceted classifications are the most widely explored of these systems and they have been proposed as suitable for online retrieval by Godert [11]; but electronic systems to manage these have not been previously described. Because documents may be included under several different nodes of a faceted classifications, the faceted classifications are a type of directed acyclic graph. On the other hand, any faceted classification can be expanded as a simple hierarchy.

Some classification systems are partially faceted. For instance, books in the DDC under Art History are organized by geographic areas and historical periods. Books organized by the Library of Congress system include Cutter number extensions which are orthogonal to the main classifications. Many other classification systems, such as the INSPEC Classification for engineering and the *ACM Computing Reviews* (CR) classification system [2] are faceted.

3. HIERARCHICAL-CLASSIFICATION INTERFACE

Figure 1 shows an interface that allows interaction with the DDC. The user has navigated the Subject Hierarchy List to **005.1000 Programming**. The interface is composed of three main groups of widgets which are described below.

3.1. Interface Widgets

3.1.1. Book Records and the Dewey Decimal Classification The DDC probably is the most widely used international classification system. It is also one of the purest hierarchies of the major library classification systems. The DDC was designed for cataloging books [18], but it has been suggested as the basis for an interface to help the casual user [16]. With the introduction of high-powered personal workstations and flexible GUIs, the accomplishment of this goal for the casual user is now feasible. The headings for a large part of the DDC were obtained and merged with the book records. While

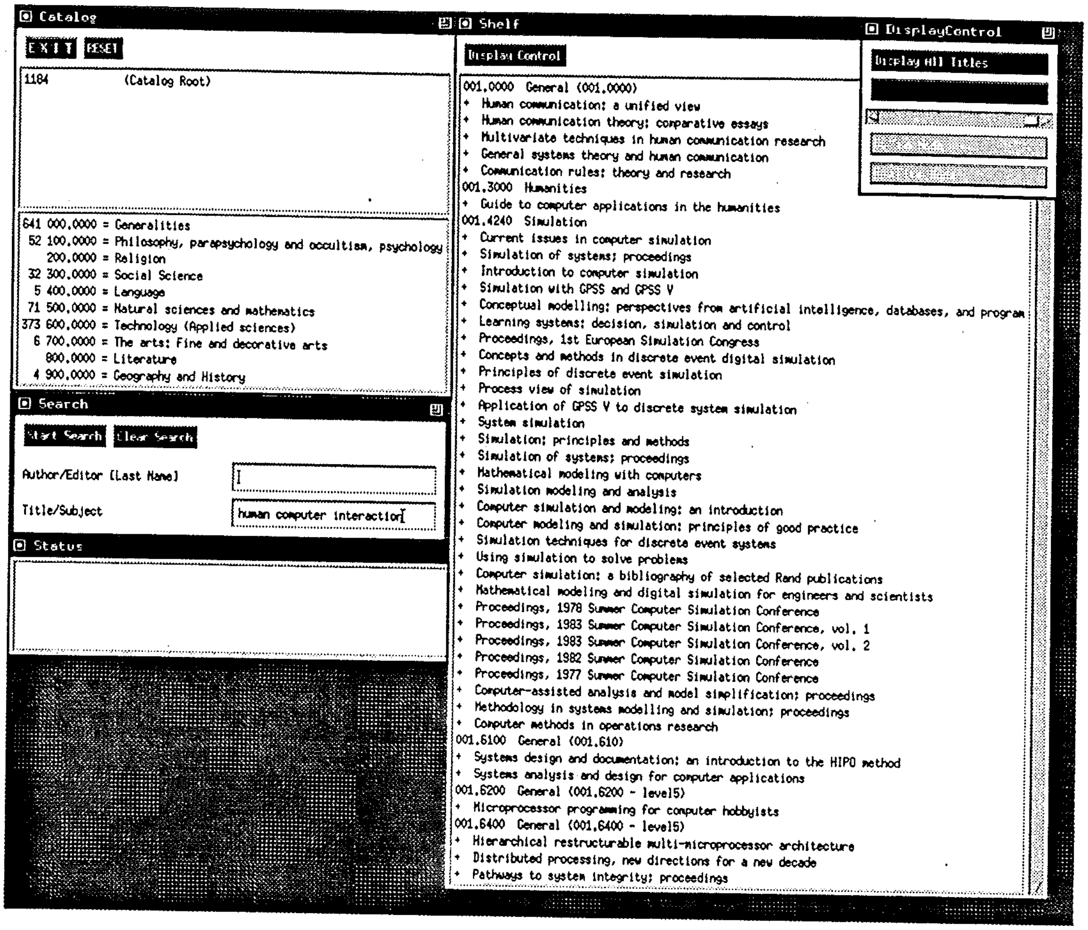


Figure 2: Interface after Search for "Human Computer Interaction".

the DDC, as with any classification system, is not suitable for all tasks, it is useful for a large range of tasks and is familiar to many users. In preparing the corpus, long call numbers were truncated to 4 decimal places. In a few cases, the hierarchy was not complete and filler headings were inserted. For instance, in the Classification immediately below the first-level node **000.0 Generalities** is the third-level node **001.0 Knowledge**. A second-level heading **000.0 General** was created to match other second-level headings under **000.0 Generalities** such as **010.0 Bibliography**.

Book and document records numbered by the DDC were obtained from the Bellcore Technical Libraries. They covered approximately 50,000 books and technical reports. Each record included the shelf number, author, title, publisher, location, a subject field, and a list of the library locations where the book was held.

3.1.2. Subject Hierarchy and Current Node Lists: The upper left quadrant of Figures 1 and 2 shows a TOC for the hierarchical interface. The TOC is split across Subject Hierarchy and Current Node Lists. Together, these widgets allow a user to navigate through the hierarchy and serve a function similar to the expandable TOC of the SuperBook browser. In a deep and wide hierarchy, such as the DDC, the contents of the expanding TOC would frequently scroll out of view. Although less information is presented in separate Subject Hierarchy and Current Node Lists than in an expanding TOC, these lists yield a more predictable display and are especially suitable for the DDC records where the shelf number provides an additional pointer into the hierarchy. Moreover, book-record hierarchies have looser semantic connections between nodes at the same level than the TOCs of most individual documents and books. Thus, displaying all choices at intermediate-level nodes would not be particularly informative.

3.1.3. Book Shelf and Book Display Widgets: The Book Shelf (right side in Figures 1 and 2) does not attempt to mimic a physical book shelf. Rather, it is a very long list of book records. The user, typically, has only a partial view of the list. The view of the Shelf is limited by the number of items that can be displayed on the screen at any time and by options that determine which book records and which attributes of those records are to be displayed.

The selection of displayed attributes is determined in response to iterative queries that control a filter mask. Thus, the Book Shelf is "dynamic" in the same sense as the dynamic graphical query interface described in [22] and as used in general purpose data viewers (e.g., [20]).

Nodes in the classification system immediately above the selected books are also presented on the Shelf. The Shelf shows nodes at different levels abutted one after the other. The default display for records on the Shelf shows titles. The user can select other attributes to be presented on the Shelf such as the author name, the length (number of paper pages), and the publisher.

When the user clicks on a book title on the Shelf, a Book Display widget opens showing the full record for that book. Indeed, it is possible to browse the Shelf by selecting successive book titles to be displayed.

3.1.4. Fielded Search Widget: The Fielded Search widget (lower left in Figure 1) generates searches on book record fields such as title, author, and subject descriptors. Three search algorithms are available: a Boolean OR of matched terms, term matches between the query and the document terms weighted by term frequencies, and Latent Semantic Indexing (LSI) [5].

For LSI [5] searches, the LSI-value for a node is derived from the position of all the terms in the book titles and subject descriptions of all the books under that node. This is conceptually similar to the approaches of [9, 13] for other search algorithms. However, it meant that individual books were not able to be located with LSI. Moreover, because the LSI searches took considerable computational resources for matching vectors, the LSI space had to be precomputed.

3.2. Browsing

The interface can be used for browsing the DDC. The Current Node List displays items that allow the user to navigate deeper into the hierarchy. Initially, the current nodes are the top-level classification terms (as shown in Figure 1). When nodes lower in the hierarchy exist, the nodes above this are marked with an "=". The Subject Hierarchy List displays the hierarchy nodes above the books currently being displayed on the Book Shelf. Clicking on one of the higher-level nodes causes the immediate descendants of the selected node to be displayed in the Current Node List. In addition, the Shelf displays books at the selected node.

3.3. Searching

Figure 2 shows the interface following a search on the terms "Human Computer Interaction". Titles that match the search are marked with a "+". In the default Hits Only display mode, the Shelf displays only the matched books and their immediate parent nodes. However, the DisplayAllTitles button (at the upper right) lets the user display all titles with the hits interspersed.

Counts of search matches are posted beside the node labels on the TOC widgets. These counts can help the user locate relevant items. For instance, in Figure 2 1184 books match the query and 641 of these are under the heading **000.0 Generalities**. This suggests that is the most promising part of the hierarchy for looking for relevant books.

The hierarchical interface is most effective for comparing documents of relatively similar retrieval values because it does not easily display quantitative information about the matches. That is, unlike typical information retrieval (IR) systems that present items ranked by a similarity metric, the interface based on hierarchical structure does not readily show graded retrieval scores. The approach taken here is to set a threshold in the ranked-ordered list and to treat all items above that threshold as hits. Initially, a titration procedure was developed to select the threshold so that, not less than 5 titles and not more than 100 titles would be presented. However, informal user testing suggested that users often wanted to override the titration setting. Thus, a slider for controlling the number of hits displayed was developed. This is similar to the use of a slider for "aggregation manipulation" [12]. In Figure 2, the slider (upper right) has been positioned to show the maximum number of hits (1184 in this example).

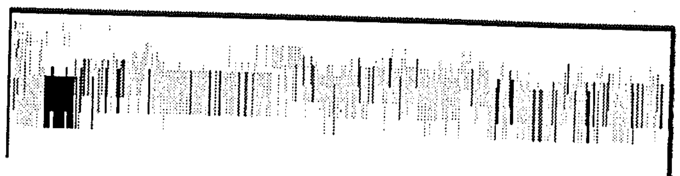


Figure 3: Graphic Display of Dewey Hierarchy after LSI Search.

It has been believed that book titles are too short to yield effective searches. However, the assumption behind this work is that there are often enough records in a node that relevant words will appear in, at least, some of them. Getting search matches on some of the titles in a node allows the user to reach that node and then to use the Shelf browsing capability of the interface and then to find the most relevant documents. In addition, following a search, the user could easily step forward and backward on the Shelf with the NextMatchNode and PreviousMatchNode buttons.

3.4. Extended Features

Several additional features were implemented for the hierarchical interface but were not included in the basic version.

3.4.1. Interactive Graphic view of DDC: Graphics can often help orient users with large amounts of data. However, graphical displays have been only lightly used in information interfaces [14]. Figure 3 shows a black-and-white view of a compressed dendrogram of the nodes in the Dewey hierarchy. Like [3], this dendrogram is interactive. In this case, clicking on the dendrogram causes the Book Shelf, Subject Hierarchy and Current Node Lists to open to the selected node. The dendrogram in Figure 3 shows search hits from an LSI search on the term "computer". Dark lines indicate better matches. Clearly, many of the computer-related books are in the early part of the hierarchy. The graphic display tool is still in early stages of development. For instance, the node representations are so closely spaced that it is difficult to see them and to select them.

3.4.2. Restricting Shelf by Attributes: Attributes, such as library location, whether the document has been checked out, and the type of document, may be used to select subsets of books controlled by menus. By selecting various library locations it is possible to examine the virtual Shelf for any one location or any combination of locations of the Bellcore Technical Libraries.

3.4.3. Additional Shelf Traversal Modes: Two additional modes for skipping through search hits on the Book Shelf were implemented. It was possible to skip by Book and by search-algorithm-match order. Specifically, the UpBook and DownBook buttons allow the user to easily find book titles that match a search. The PreviousBookInOrder and NextBookInOrder buttons let the user examine books in the ranked order in which they matched the query. However, it is easy for the user to lose orientation because the books are not necessarily in order and the user viewing them may jump around the hierarchy. If the user requests NextBookInOrder after all the books in the initial set have been viewed, the set expands by relaxing the threshold.

3.4.4. Similar Books: An option for the Book Display allows the user to request Similar Books. It searches for books similar to the displayed book where similarity is determined by one of the retrieval algorithms rather than by shelf proximity. This option spawns a new search that, when it follows an initial search, is a type of relevance feedback. Because the book records are short, the Similar Book requests yield some spurious matches. As with the initial searches, posting similar-book hits against the Subject Hierarchy List allows the user to follow the classification semantics to identify relevant items. The Book Display also contains options for presenting other books by the same author. This links books across leaf nodes of the hierarchy.

3.4.5. Lateral Links: For especially complex hierarchies, when a person using the browser reaches a terminal node they may not find exactly the information they are looking for but they may suspect they are close to it. Requesting a search for Similar Books (see above) would be one way to find other relevant sections of the hierarchy, but it is also possible to have precomputed lateral links between nodes (i.e., "distributed relatives"). A mechanism was implemented for this, in which a button was associated with each node and clicking on that button presented a list of other related nodes. At some point, these complex hierarchies would be better represented by faceted classification systems (see Section 3).

3.4.6. User Restricted Collections: In many cases, a user would be willing to restrict searches to certain segments of the classification hierarchy. This could improve computational efficiency and would focus the users attention. While that capability was not essential for the current prototype with about 50K book records on a powerful workstation, for a much larger collection (e.g., for the Library of Congress collection or for World-Wide Web (WWW) pages on the Internet) the user should be able to specify subsets of the records to search. For this system, users sub-selected nodes to include on a separate shelf and they could toggle back and forth to that shelf.

4. INTERFACE FOR FACETED CLASSIFICATIONS

Figure 3 shows an interface for browsing the computer science literature by means of the *Computing Reviews* classification. The test corpus consisted of doctoral dissertations cited in *ACM Computing Archive* [1] as published in 1992. The key idea is selection by specifying multiple constraints. Of course, there is no linear organization of documents for display in this collection; thus, the order of the nodes in the shelf displays is undetermined.

4.1. Interface Widgets

4.1.1. Cascading Facet Menus and Active Constraints Widget: Major categories are chosen from the Facets widget at the upper left of Figure 4. These selections open cascaded menus that display lower-level categories. When the "+" to the right of the facet label is selected, the facet is added to the Current Constraint List (left middle in Figure 4).

To show the context of the selected constraint labels, the parents of the constraints are displayed in parentheses on the Constraint List. The Shelf is updated with articles that match the constraints. Of course, the constraints propagate to all their descendants. Constraints can be dropped from the Constraint List by clicking on the "–" on the right side of the widget.

The interface allows the user either to take documents that match the union of the constraints (AND) or the intersection of the constraints (OR). For large collections, there are often far too many matches for the union. By switching to the AND display, the most relevant documents can be easily found. For the ACM CR collection, there is substantial variability in the number of categories assigned and the criteria for determining relevance of those categories.

4.1.2. Shelf: Because most of the documents are assigned to several categories, a user could find a relevant node and then find other nodes that have similar classifications. The overlapping categories are presented in the current interface by selecting the "o" from the first vector on the right side of the Facet Menu widget.

Among doctoral dissertations that were cited in *ACM Computing Archive* [1] as published in 1992, the categories that had two or more overlaps to **H.3.3 Information Storage and Retrieval** were **H.2.4 Systems**, **H.2.0 General**, **D.3.2 Design Styles**, **H.5.2 User Interfaces**, and **I.2.6 Learning**. Thus, a user who accessed articles under **H.3.3** could examine those other categories for relevant material. This is a type of *lateral link* across the hierarchy (see "Extended Features" section above).

4.1.3. Searches: Currently, term-frequency weighted searches are implemented in this interface. In one mode, it is possible to ask for all document titles to be included

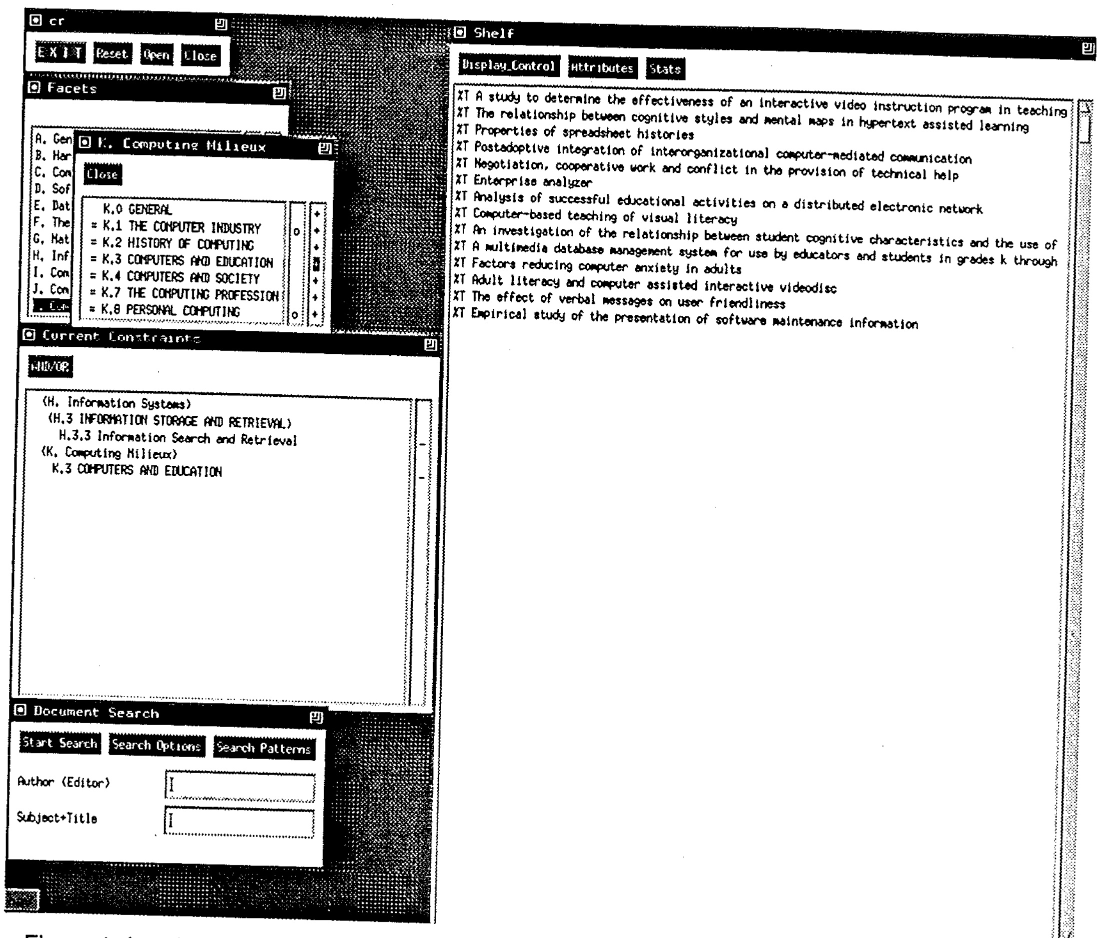


Figure 4: Interface for *Computing Reviews* Classification with Two Constraints Selected.

in the search. It is also possible to limit the search to those documents that match the constraints.

Posting search hits against the hierarchy is more complicated in this case than for the simple hierarchical display because a single document can belong to several categories. The current system uses fractional category memberships when the hits are spread across categories. As noted above, the Book Shelf for the facet interface has no a priori order. Thus, there is no natural order to display search hits. On the other hand, a variety of other ad hoc organizations are possible. For instance, the categories might be ordered by the density of hits. A related problem is which facet hierarchy to pop-open after a search (perhaps to help guide the user to further refine the search).

5. DISCUSSION

5.1. User Studies

While formal user studies have not been conducted on these interfaces, informal feedback from users of the hierarchical interface has been generally favorable. One major innovation here has been the introduction of a Book Shelf. Because this is the only full-scale system to include a Book Shelf (and hence the only system to allow browsing of books by shelf order), it is not clear what sort of evaluation is most reasonable.

The greatest problem with these interfaces appears to be complex interactions among features. For instance, in the Hits Only mode there are often too few selections to fill the Shelf Display; thus, the UpBook and DownBook buttons have no effect. In addition, some test users have suggested that the elision in the Hits Only mode should apply to the TOC as well as the Book Shelf. Completely shifting context from one set of screens to another (e.g., with the similar books option) is also difficult.

Beyond the problems of the interface design, there are limitations inherent in this type of interface for hierarchical classification systems. A substantial concern is the user does not know how many books are included under each node. For parts of the hierarchy hierarchy, a user may know or may be able to take a good guess; however, the user may not be at all familiar with other parts of the hierarchy.

The facet interface is probably harder to use than the simple hierarchical interface. This is because of the complexity of managing multiple facet hierarchies and the lack of a natural shelf order for the documents. Moreover, the facet interface described here has not been as well developed as the simple hierarchical interface. For instance, graphical displays might be especially useful for navigation of the facet hierarchies.

5.2. Integration with Other Information Systems

These interfaces could provide the basis for access to additional electronic information sources. Clearly, it would be possible to have the short document records pointing to the full text of the books and documents. Moreover, encyclopedia articles describing authors could easily be presented. Likewise, book reviews, citation statistics, circulation data, and user annotations could be included as part of the Book Display. Conversely, an electronic encyclopedia could access the OPAC for bibliographies.

Overall, these interfaces suggest that the structure of a classification system can be a useful aid for searching and navigating a digital library. Indeed, it may be worth exploring how digital library classifications can be extended to finding information in less structured domains such as for information in the WWW.

5.3. Envoi

Techniques such as the PreviousMatchNode/NextMatchNode buttons and lateral linking show how search-based IR and structure-based Hypertext approaches can be combined. It is also worth noting that structure could be used to enhance a search-based OPAC (e.g., [8]). In any event, while the DDC provides links to related documents, there are many other dimensions of similarity (e.g., author, citations, publisher) that could be used for linking as well. It remains to be seen whether these dimensions can be coordinated into useful interfaces.

ACKNOWLEDGMENTS

The DDC was used with the permission of the Online Computer Library Center (OCLC). The collection of book records used here was developed for test purposes and is not a Bellcore product. A much earlier version of this paper appeared in *Digital Libraries'94*, College Station, TX, June, 1994.

REFERENCES

1. ACM, *ACM Computing Archive*, 1994, New York.
2. ACM, ACM Computing Reviews Classification System. *ACM Computing Reviews 35* (1994) 4-44.
3. Allen, R.B., Obry, P., and Littman, M., An Interface for Navigating Clustered Document Sets Returned by Queries. *Proceedings of SIGOIS* (Milpitas, CA, June) ACM, New York, 1993, 203-208.
4. Borgman, C.L., Walter, V.A., Rosenberg, J.B., and Gallagher, A.L., Children's Use of a Direct Manipulation Library Catalog. *ACM SIGCHI Bulletin 23*, 4(Oct. 1991) 69-70.
5. Deerwester, S., Dumais, S., Furnas, G., Landauer, T.K., and Harshman, R., Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science 41* (1990), 391-407.
6. Egan, D., Lesk, M.E., Ketchum, D., Lochbaum, C.C., Remde, J.R., and Landauer, T.K., Hypertext for the Electronic Library? CORE Sample Results. *Hypertext '89* (Pittsburgh, Nov.) ACM, New York, 1989, 299-312.
7. Egan, D., Remde, J.R., Gomez, L.M., Landauer, T.K., Eberhardt, J., and Lochbaum, C.C., Formative Design and Evaluation of SuperBook. *ACM Transactions on Information Systems 7* (1989) 30-57.
8. Fox, E.A., France, R.K., Sahle, E., Daoud, A., and Cline, B.E., Development of a Modern OPAC: From REVTOLC to MARIAN. *Proceedings of SIGIR'93* (Pittsburgh, June) ACM, New York, 1993, 248-259.
9. Frisse, M.E., Cousins, S.B., and Hassan, S., WALT: A Research Environment for Medical Hypertext. *Hypertext'92* (San Antonio, Nov.) ACM, New York, 1992, 389-394.
10. Furnas, G.W. and Zacks, J., Multitrees: Enriching and Reusing Hierarchical Structure. *ACM SIGCHI'93* (Boston, Apr.), ACM, New York, 1993, 330-336.
11. Godert, W., Facet Classification in Online Retrieval. *International Classification 18* (1991) 98-109.
12. Goldstein, J. and Roth, S.F., Using Aggregation and Dynamic Queries for Exploring Large Data Sets. *ACM SIGCHI'93* (Boston, Apr.), ACM, New York, 1993, 23-29.
13. Hearst, M. and Plaunt, C., Subtopic Structuring for Full-length Document Access. *Proceedings SIGIR'93* (Pittsburgh, June), ACM, New York, 1993, 59-68.
14. Lesk, M.E., What To Do When There's Too Much Information? *Hypertext '89* (Pittsburgh, Nov.) ACM, New York, 1989, 305-318.
15. Mann, T., *Library Research Models*, New York, Oxford University Press, 1993.
16. Markey, K. and Demeyer, A.N., *Dewey Decimal Classification Online Project: Evaluation of Library Schedule and Index Integrated into the Subject Searching Capabilities of an Online Catalog*, OCLC, Dublin OH, 1986, OPR/RR-86-1.
17. Micco, M. and Basista, T., Beyond Subject Access: The Next Generation of OPAC Software. *Proceedings Integrated Online Library Systems* (1991), 103-112.
18. OCLC (Forrest Press), *Electronic Dewey*. Dublin OH, 1993.
19. Pejtersen, A.M., A Library System for Information Retrieval Based on a Cognitive Task Analysis and Supported by an Icon-Based Interface. *Proceedings of SIGIR'89* (Cambridge, MA, June) ACM, New York, 1989, 40-47.

20. Swayne, D.F., Cook, D., and Buja, A., Interactive Dynamic Graphics in the Xwindow System with a Link to S. *Proceedings of the Section on Statistical Graphics of the American Statistical Association* (Atlanta) ASA, 1991, 1-8.

21. Vickery, B.C., *Faceted Classification.* New Brunswick, NJ, Rutgers University Press, 1965.

22. Williamson, C. and Shneiderman, B., The Dynamic HomeFinder: Evaluating Dynamic Queries in a Real-Estate Information Exploration System. *Proceedings of SIGIR'92* (Copenhagen, June) ACM, New York, 1992, 338-346.

Modeling for Interaction in Virtual Worlds

Curtis Lisle,
University of Central Florida

Keywords: virtual world, virtual reality, modeling, software architecture

1.0 Abstract

An effective story gets the reader involved--hoping or worrying about how the story will finally turn out. Virtual Reality (VR) technology holds promise as a powerful new medium for story-telling as well as other types of communication. However many of today's products have not met the expectations of either researchers or end users. We present the case that effective participation by a human in a virtual world depends on robust computer data structures to support behaviors and interaction. The modeling approach to the virtual world should drive the architecture of the computer simulation. In this paper we discuss issues in modeling virtual worlds for interaction and suggest a software architecture for such simulations.

2.0 Introduction

Immersion is a subjective measure of the amount of "belief in the experience" that a human participant has while in a VR [1]. We believe that the interaction afforded by a particular VR experience plays a major role in achieving the immersion effect for its participants. To become an effective medium, VR requires rich modeling capability to support better interaction. What if I want to read one of the books I find in a VR? Will the VR system allow me to?

Over recent years, much of the research efforts in VR simulation systems have focused on the human/computer interface and rendering technologies. While these areas are important, the capability and effectiveness of a VR system is also dependent upon the computer modeling of the virtual world and the architecture of the simulation software. For example, how multiple workstations synchronize and maintain a single, shared virtual environment for all participants.

3.0 Rendering is Not the Problem Anymore

VR technology has come out of the interactive computer graphics community where rendering data structures and techniques were the first focus of research. However, the high-quality, high-speed rendering available today makes us want to interact with the images we can now render. The applications that VR technology faces today (e.g. electronic publishing, entertainment, interactive graphics in education, computer modeling of physical or biological processes.) are substantially different problems than those of early graphics research.

As VR technology matures, it should draw upon the lessons learned by the military simulation and training community. Today's military flight simulators and ground vehicle simulators are mature virtual world simulation systems in many respects, but they primarily use rendering-based data structures which limits the quality of the simulation [3]. Polygonal representations of the objects in a simulated environment support rendering well but are *semantically impoverished* for the task of virtual world modeling. Consider a car driving over rolling terrain: we are accustomed to the motion of a vehicle as the wheels pass over small bumps. But if the ground elevations were represented by a triangular-irregular networks (TINs)[4] , as in the majority of training databases today, the vehicle would be *terrain following* over artificial seams between triangles .

Experience in vehicle dynamics models indicates that as the vehicle's dynamics improves, it requires more sophisticated environmental models to achieved the desired behavioral realism [5].

Instead of focusing only on rendering data structures, we feel that research is also needed in the remaining areas:

<u>An On-line Data Structure</u> - Interaction requires a data structure to support the types of queries and changes which are appropriate for VR applications. The data structure should be capable of real-time updates as the modeled environment is affected. We believe the most difficult challenge here is simultaneous support of participants operating at different levels-of-fidelity. For example, one participant could be closely examining the soil in a valley while another is flying high overhead.

<u>Support for a Shared Experience</u> - Multiple participants should be able to be in a virtual world simultaneously. Each participant has a view of the environment which is appropriate from his or her viewpoint. However, the environment model should be able to support all simultaneously. This type of application have been predicted by the leaders in the database research community [6].

<u>A Database Interface Protocol</u> - If different applications are running on a shared database of a virtual world, a protocol for interaction must be defined. The protocol would describe how to query and effect objects and how to control an application's view of the database.

4.0 A Protocol for Interaction

The military training community has developed the Distributed Interactive Simulation protocol for use when multiple vehicle simulators are connected [7]. The protocol uses UDP/IP packets for communication and operates on a shared network with no central server. Packets contain vehicle state information or notification of events affecting one or more of the vehicles in the simulation. The messages in this protocol make it specific for use in synthetic battlefield simulation, but the DIS concept of *dead reckoning* applies to all types of virtual environments. In dead reckoning, all entities in the environment send out messages only when something in their state changes greater than a predetermined allowable threshold. With this approach, network traffic is reduced but all participants are informed of any major changes.

DIS network messages use a standardized data format [7]. However, there is more to a protocol design for object interaction than the message format and the low-level communication mechanism. For a VR protocol to be successful, the semantics of the connected simulations must be compatible. Otherwise protocol messages will be misinterpreted. We believe that semantics issues are the area where research is most needed.

Modeling the virtual world using Object-Oriented software design techniques causes the software architecture to reflect the relationships of the world it is simulating [2]. In other words, the form of the software follows patterns in the modeled environment. Consider the example of simulating a billiards table using instances of the hypothetical software classes *BilliardsTable* and *BilliardsBall*. During the simulation, messages are sent between the objects indicating collisions and changes in object position. Creating software classes which reflect virtual objects results in a more intuitive design for both the developer and end users. This modeling approach has been tested in the *PM system* [8]. In the PM system, a protocol was developed for physical objects and constraints which allowed the simulation of simple sets of objects interacting without precalculated behavior functions. Interaction was accomplished through messages between the objects. In this case, *the interaction protocol is*

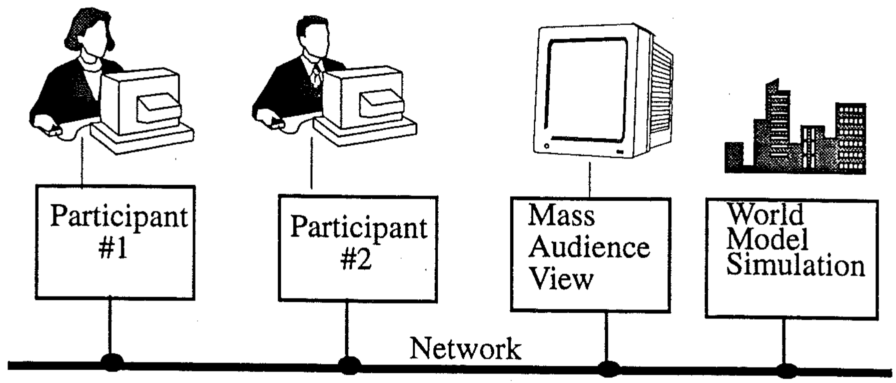


Figure 1: A Multi-Participant VR System

defined by the interfaces to the object classes themselves.

The Hypertext Markup Language (HTML) has proven to be a very-effective representation for the creation, storage, reference, and transfer of hypertext documents. With the use of a tool like *Mosaic*, HTML documents give the user the ability to browse and interact with a document in a way still envied by virtual world modelers. The question raised earlier, "What if I want to read a book I find in the virtual world?" can now be answered, "Open your HTML browser on the book." To generalize this example, *virtual objects should be represented using data structures which can be exploited by tools available to the VR participant.*

Work has already begun to extend the concept of HTML for representing virtual worlds, and the experimental standard is called VRML for *Virtual Reality Modeling Language* [9]. We believe that standards such as these could eventually help virtual world builders to share a common modeling language, interaction protocol, and maybe even world-building tools.

5.0 A Shared Environment

Consider an example VR application shown in Figure 1. It has two immersed participants, a large screen (like a magic carpet) for observers, and a server to model the behavior of objects in the world. Several different applications are all sharing a common world model. Each application is running on a separate computer connected via a local-area network.

For this VR system to be realized, a shared virtual environment should exist such that each computer on the network can participate as appropriate for its individual application. The primary technical obstacle here is the maintenance, distribution, and concurrency control of the shared environment. Luckily, Object Oriented Database technology is just arriving to aid in the solution of this problem.

6.0 A Virtual World Simulation Architecture

In light of the issues raised earlier in the paper, let's consider the example presented in the last section as it would be handled using a system architecture like that of Figure 2. Running on each computer, an Object-Oriented Database (OODB) would manage the storage, retrieval, and distribution of a consistent set of objects across the set of computers. On top of the OODB, class definitions which defined the virtual objects and the protocol for interaction would serve as a view of the shared virtual world. With this approach, the application programs manipulate the objects in the virtual world at a more intuitive, abstract level. We

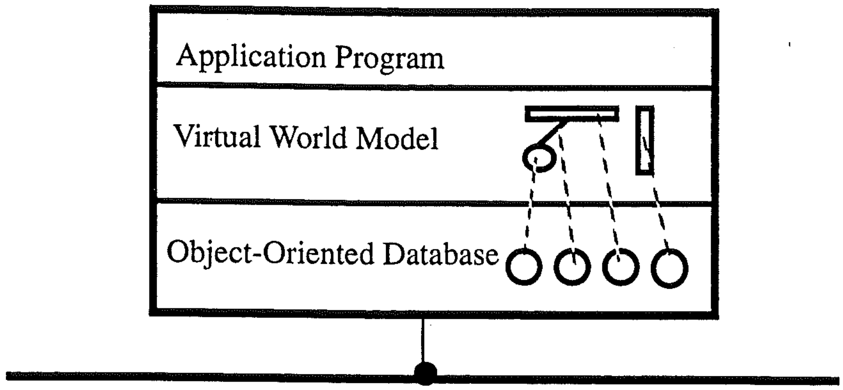


Figure 2: A Proposed VR Simulation Architecture

believe that this layer of software abstraction will make VR programming conceptually cleaner for only a modest performance penalty. The management of distributed objects is already supported by several commercial OODB products [10] [11].

Another way of describing this approach is that the application program would be designed to make calls to a *Virtual World API* (Application Programmer's Interface). The API would support the actions using the VR protocol described previously. Objects instanced in the virtual world would be stored in the Object-Oriented Database to provide persistence and distribution across platforms on the common network. Consider how the scenario presented in Figure 1 would be constructed using the architecture suggested above. In Figure3, the shared environment is implemented through the use of the OODB and *Virtual World Model* layers. Different application programs on each computer would only interact with the Virtual World Model layer as required for their application. In this example, the two participant simulations would each be able to change the state of the world through their input devices while the Magic Carpet view only traverses and draws objects. The Environment State simulation calculates any reaction the environment has to the actions of the participants. An architecture similar to this is developed and discussed in [3].

7.0 Conclusion and Future Work

We believe the software architecture for a VR simulation system should follow the protocol used for representation and communication of the objects in the virtual world. The modeling approach (what data abstractions exist and how they are used) has a profound impact on the software design. Research is needed to support development of a hierarchy of classes where objects belonging to each class interact with the virtual world using a common protocol. Database technology will also play a key role in the development of future VR systems since the database system will handle the storage and manipulation of virtual objects. Object Oriented Database technology coupled with experimental protocols under development in projects like TSIMMIS [13] hold promise.

8.0 Bibliography

[1] B. G. Witmer, "Measuring Presence In Virtual Environments", *Technical Report, U.S. Army Research Institute for the Behavioral and Social Sciences*, April 1994.

[2] J. Burg et al., "Behavioral Representation in Virtual Reality", *Proceedings of the 1st Annual Behavioral Representation Conference*, Orlando, FL, April 1991.

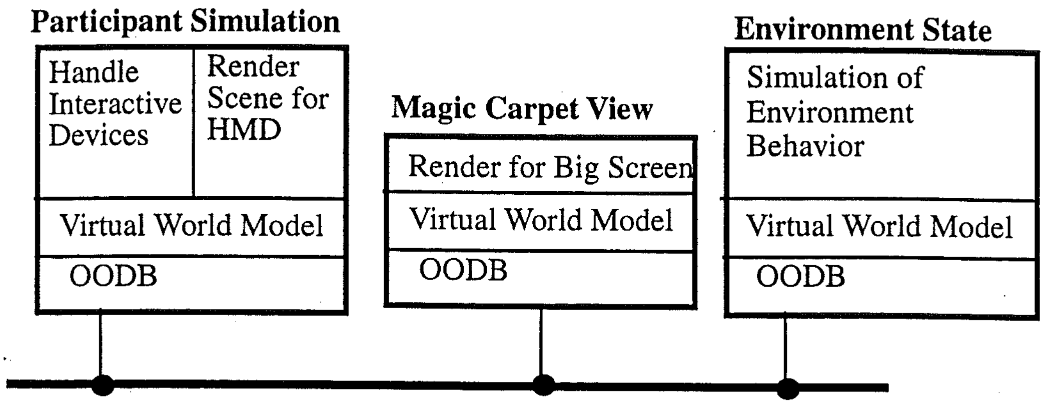


Figure 3: A Multi-Participant VR System

[3] C. Lisle et al., "Architectures for Dynamic Terrain and Dynamic Environments", *Proceedings of the 10th Workshop for the Interoperation of Defence Simulations*, Institute for Simulation and Training, Orlando, FL, March, 1994. (Available via http://www.vsl.ist.ucf/~deg/papers/DISMarch94PositionPaper.ps.gz)

[4] L. Scarlatos, "Spatial Data Representations for Rapid Visualization and Analysis", Ph.D. Dissertation, Computer Science, State University of New York at Stony Brook, 1993.

[5] G. Prasad and M. Altman, "Concerns Over Adding Vehicle Mobility Calculations to DIS Exercises", *Proceedings of the 12th Workshop for the Interoperation of Defence Simulations*, Institute for Simulation and Training, Orlando, FL, March, 1995.

[6] Silberschatz, Ullman, Stonebraker, "Database Systems: Achievements and Opportunities", *Communications of the ACM*, 34(10): 110-120. 1991.

[7] *DIS Operational Concept 2.3*, IST-93-25, Institute for Simulation and Training, University of Central Florida, Orlando, FL, pg. 4

[8] C. Lisle and J. M. Moshell, "Object-Oriented Physical Modelling", *Journal of Systems Engineering(1993):3:191-201*, Springer-Verlag, London.

[9] VRML WWW home page: http://vrml.wired.com; "VRML 1.0 Specification", available via WWW at http://www.eit.com/vrml/vrmlspec.html

[10] *Objectstore Technical Overview V2.0*, Object Design, Inc.,One New England Executive Park, Burlington, MA 01803, July 1992.

[11] *Objectivity/DB Technical Overview V2.0*, Objectivity, Inc., 800 El Camino Real, Menlo Park, CA 94025, March 1993

[12] J. M. Moshell et al., "Dynamic Terrain", *Simulation 62:1*, Simulation Councils, Inc., January 1994, pg.29-40.

[13] S. Chawathe et al., "The TSIMMIS Project: Integration of Heterogeneous Information Sources," *Proceedings of the IPSJ Conference*, Tokyo, Japan, October 1994.

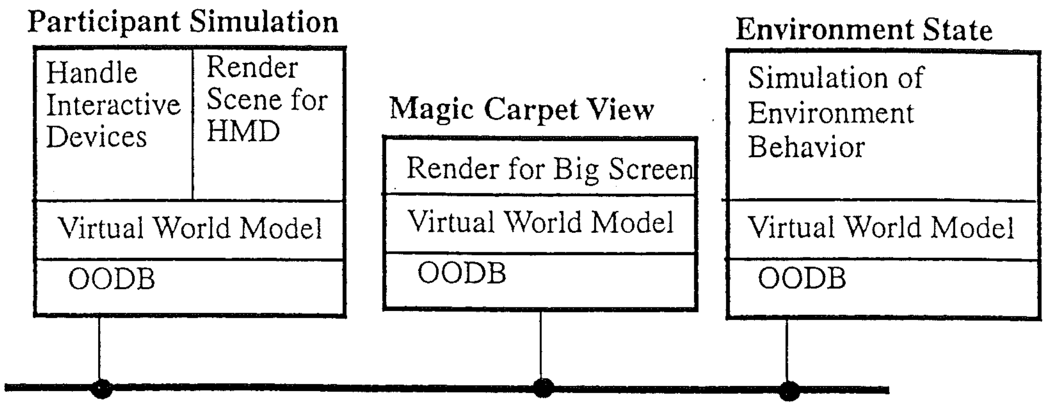


Figure 3: A Multi-Participant VR System

[3] C. Lisle et al., "Architectures for Dynamic Terrain and Dynamic Environments", *Proceedings of the 10th Workshop for the Interoperation of Defence Simulations*, Institute for Simulation and Training, Orlando, FL, March, 1994. (Available via http://www.vsl.ist.ucf/~deg/papers/DISMarch94PositionPaper.ps.gz)

[4] L. Scarlatos, "Spatial Data Representations for Rapid Visualization and Analysis", Ph.D. Dissertation, Computer Science, State University of New York at Stony Brook, 1993.

[5] G. Prasad and M. Altman, "Concerns Over Adding Vehicle Mobility Calculations to DIS Exercises", *Proceedings of the 12th Workshop for the Interoperation of Defence Simulations*, Institute for Simulation and Training, Orlando, FL, March, 1995.

[6] Silberschatz, Ullman, Stonebraker, "Database Systems: Achievements and Opportunities", *Communications of the ACM*, 34(10): 110-120. 1991.

[7] *DIS Operational Concept 2.3*, IST-93-25, Institute for Simulation and Training, University of Central Florida, Orlando, FL, pg. 4

[8] C. Lisle and J. M. Moshell, "Object-Oriented Physical Modelling", *Journal of Systems Engineering(1993):3:191-201*, Springer-Verlag, London.

[9] VRML WWW home page: http://vrml.wired.com; "VRML 1.0 Specification", available via WWW at http://www.eit.com/vrml/vrmlspec.html

[10] *Objectstore Technical Overview V2.0*, Object Design, Inc.,One New England Executive Park, Burlington, MA 01803, July 1992.

[11] *Objectivity/DB Technical Overview V2.0*, Objectivity, Inc., 800 El Camino Real, Menlo Park, CA 94025, March 1993

[12] J. M. Moshell et al., "Dynamic Terrain", *Simulation 62:1*, Simulation Councils, Inc., January 1994, pg.29-40.

[13] S. Chawathe et al., "The TSIMMIS Project: Integration of Heterogeneous Information Sources," *Proceedings of the IPSJ Conference*, Tokyo, Japan, October 1994.

Direct Metaphor and User Interaction in the Electronic Libraries of the Future

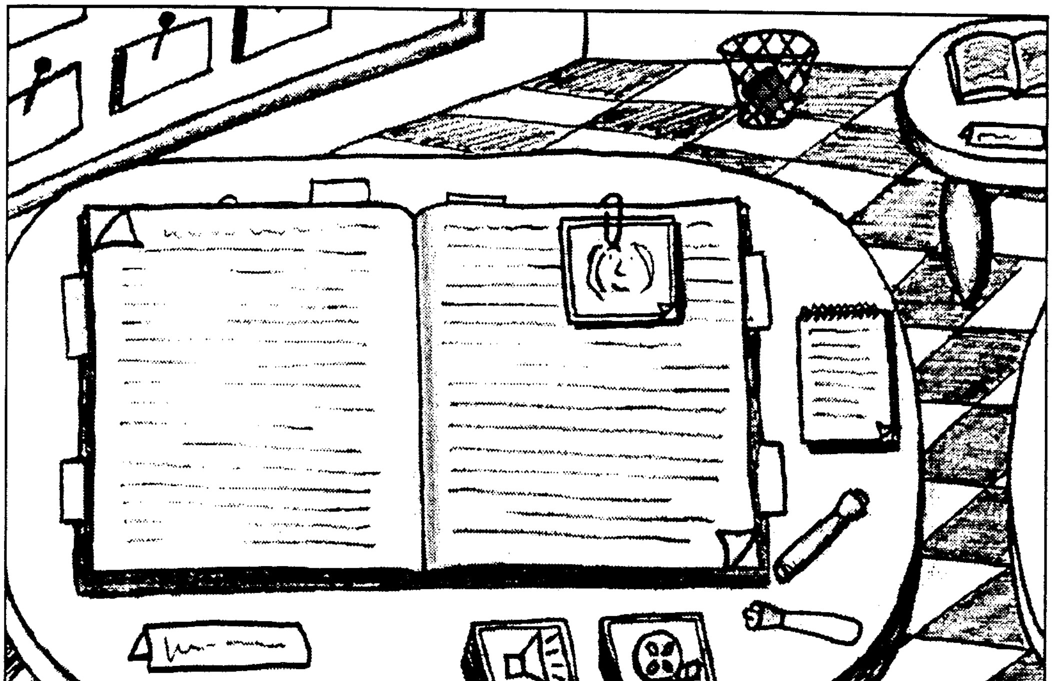
By Matthew Owen Williams
<mowillia@us.oracle.com>

Part 1. Sally's Research Paper

As Sally Rogers leafs through the latest on-line issue of *International Geographic* in her San Francisco townhouse, she hears the chime that signals an incoming video conference; it's her history professor, Laura Donnelly, calling from her offices in Hawaii. Professor Donnelley is calling about Sally's term paper, an abstract of which Sally e-mailed to her professor last week. Sally opens up her project notebook by locating it on her computer desktop and takes a few additional notes during the conversation. The professor approves Sally's thesis paragraph and, before signing off, they set a date for their next videocon, two weeks in the future.

Sally then enters the virtual room she created specifically for this project through a door in her personal library. She is presented with a number of tables bearing piles of books. In addition to the pair of tables with research she has already performed to come up with her thesis, a new table is present with a stack of books that her professor recommended as well as a few notebooks her professor has lent her. Sally navigates around this virtual space, examining the different tables and rearranging the piles of books on each table as she wishes. She runs her thesis paragraph through a lexical analysis program and uses the results to run a search that generates another table full of related books. A few other queries bring the total up to eight research topic tables.

Digging in, she touches a book and it zooms forward, filling 75% of the screen, with the rest of the space showing the table the book was on along with parts of the rest of the library. *(fig. 1)*



(fig. 1)

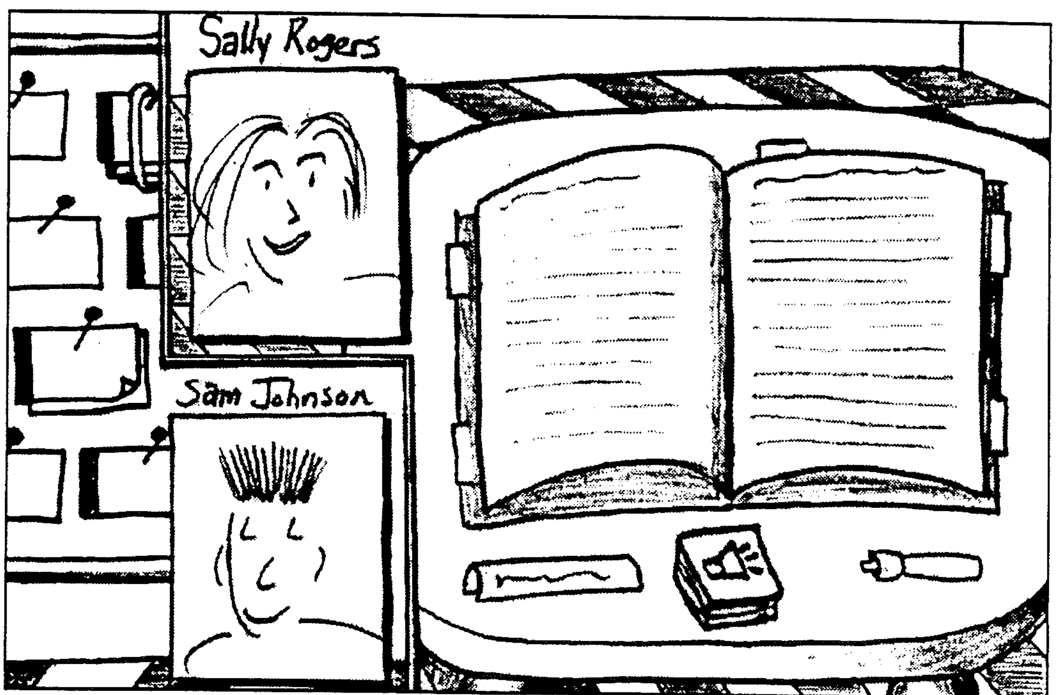
After admiring the cover art, she advances to the table of contents where the lexical analysis she ran earlier has highlighted certain parts of the book that are relevant to her topic. Tapping on an entry in the table of contents flips to that page and she begins reading. She advances through the book at her own pace, occasionally instructing the computer to read a short passage

aloud so she can hear how it sounds or following a footnote link to another work.

Also on the table are a number of notebooks in various colors, each representing a particular research topic. When she finds a section that is relevant to a particular topic, she pulls a color coded highlighter pen from the table-top and marks that section in the book. This creates a new page in the correspondingly colored notebook, transcribing the highlighted quote as well leaving space for her to take additional notes. At one point she finds a certain passage in the book particularly interesting, but she doesn't have time to follow up on it now. She rips a page from the sticky notepad on her desk marked with a speaker. An interactive agent then appears to whom she dictates her thoughts; he transcribes her speech and leaves both the audio and the text on a small sheet of paper. She then attaches this sticky note both to the source text as well as to the appropriate notebook on the tabletop.

As she leaves each page, the date and time when she first turned to the page is automatically stamped lightly in an unobtrusive place on the bottom of the page. If she returns to this page later, touching the time stamp will bring up a menu of the various times she has read that particular passage. Selecting one of these times allows her to go back through the research thread she followed in that pass through the book. Any links that she followed before will be marked so that she can easily retrace her steps, although she is of course able to strike off in a new direction at any time.

The next book she chooses is one that her professor recommended. Annotated sections are already highlighted in yet another color. Touching an annotation brings up the relevant page in the notebook her professor lent her. In addition to the highlighting, her professor has also left a few video notes in the book. These notes are shown as a small picture of her professor paper clipped to the source text and to the relevant notebook pages. Sally can listen to her professor's comments by merely tapping on the picture. Hypertext links, represented as footnotes in the professor's notebook, give Sally valuable leads on related journal and magazine articles. If she doesn't agree that a particular book is relevant to her thesis, Sally can choose to look only at the sections that her professor has marked in the book, skipping the unhighlighted sections.



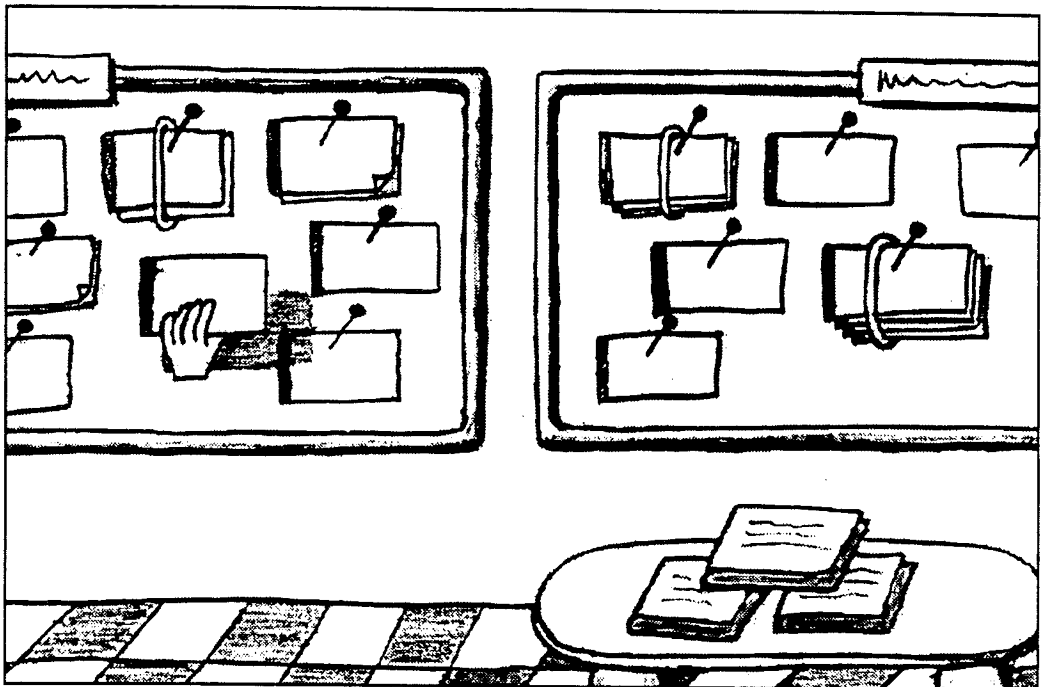
(fig. 2)

As Sally makes her way through the piles of books in her research library, books that she doesn't find particularly interesting are dumped into the dustbin and other books are sorted and piled along with relevant notebooks on her research tables. If she changes her mind about a book she has thrown away, she can look through her wastebasket which is presented as a timeline history of books that she has read and rejected. Locating the book she wants to

retrieve, she drags it back onto the appropriate table-top and opens it up again.

In her spare time, Sally is a member of a history student's chat group that meets twice a week on-line. Through this forum she has met Sam Johnson, a masters candidate at the University of Colorado, who is working up a paper on a similar topic. She places a videocon call to Sam and visits him in his personal library. In addition to seeing the transmitted images of themselves and each other, for the duration of the call their computer screens are split between the two libraries. *(fig. 2)* They each control which part of their library the other will see, although each can choose how much of the other's library will occupy their screen by dragging the split bar to one side or the other. Thus they can both look at a particular book in one library as well as easily copying books and notes back and forth between the virtual spaces by dragging from tabletop to tabletop. Sally thanks Sam for the journal articles that he has shared with her and signs off, promising to send him a copy of her paper when it is completed.

After working her way through most of the books on each table, she decides it is time to get organized. She drags a notebook up to a blank cork board on the wall, transforming the notebook into a set of index cards pinned to the board. *(fig. 3)* The highlight color for the notebook is indicated by a colored band on the edge of the cards. She can move the notes around on the board in any order she wants,



(fig. 3)

rearranging and duplicating notes between different boards as she pleases. Dropping one notecard on top of another places a rubber band around them and links them together; from then on they move around as a unit. She can also annotate an entire group of cards by attaching a sticky note to the group.

She next converts each ordered set of note cards into outline page, represented as loose sheets of hole-punched paper. She strings the outlines together by placing them in a three ring binder on her desk which automatically expands to accommodate the new text. She can switch to a plain text version of the binder, represented as a book, to view quotes, add to her notes, make links to specific sources, etc. This outline will later become the table of contents for her paper.

Once her outline is assembled she starts actually writing the paper. She expands on the outlined topics by dictating to a secretary agent, who transcribes the speech and inserts it into the appropriate place in the paper. She pulls pictures and quotes from her notebooks and source texts into her evolving book which automatically incorporates these graphical elements and inserts live footnotes in the appropriate places. At any point she can switch back to the notecard view, jump to the origin of a quote by tapping on it, or make new notes, paper clipping them to the text of her book. A live bibliography is automatically generated from her note sources,

displayed both in the traditional style and as a series of book covers; she can add books that she didn't explicitly reference to the bibliography by simply dragging them in.

When she feels it is ready, she sends a live draft (automatically updated as she changes the original) to her professor by dropping it on professor Donnelley's picture, attached to the front of her address book. As her professor reads the paper, her comments appear in red ink overlaid in Sally's version. If Sally agrees with her professor's remarks, she modifies the text in whatever form is most natural - simple edits may be accomplished with the work in book form but for rearranging sections she may go back to the outline view or she might even switch to the notecard view if she wants to add an entirely new section. Sally can then remove her professor's remarks by merely erasing them. They go through a few times revision cycles as the work evolves from a first draft to a final paper, holding occasional videocons to discuss an important point "in person." When both are satisfied with the work, the new essay is published and officially added to the public library, and Sally gets an 'A' in the course.

Part 2. Discourse

Although the above story may sound like science fiction, most of it can be implemented today with current software technology. For example, systems for taking video and audio annotations already exist, though the most sophisticated and thus most useful of these systems are out of the reach of the average student. One can assume that the passage of time will remedy this situation, bringing the technology into everyone's hands. There are, however, three key components of enabling this technology as it is outlined above that have not been fully explored in current electronic library systems:

I. direct metaphor as interface,

II. context-based polymorphism of information, and

III. a time-based memory of user interaction with the printed word.

I. Direct metaphor as interface

The use of direct metaphor as interface is not new; it is the underlying concept behind first the Xerox Star and later the Macintosh line of personal computers. It now, in some form or other, characterizes user interaction with the file system of most graphical user interface operating systems. Shuffling icons around, organizing them in "folders," throwing things away by placing them in the "trash," etc. is now standard procedure and people seem very comfortable with the concepts behind, for example, the Macintosh Finder. However, once the user leaves the file system, the real world metaphor is almost always lost; word processors, characterized as endlessly scrolling columns of text controlled by a vertical scroll bar, bear very little relation to a traditional book or magazine. If people are to truly embrace reading and researching on a computer, interaction through direct metaphor needs to be carried out through the entire process.

The interface of General Magic's Magic Cap personal communicators is a huge leap in this direction; every interaction with the system takes place in some real-world context, from the user's desk to a library with shelves of books to a downtown area for interacting with the outside world. Although this "virtual reality" is simply illustrated, the user is quickly drawn into this pleasant little virtual world precisely because it is so similar to the physical world we inhabit now. Concepts that are complex or confusing in traditional software applications are rendered simple because the user brings so many physical world concepts into this virtual world. Interaction with virtual agents - small, specialized software programs that execute tasks that would be tedious to do by hand - is the final piece of the puzzle that makes actions such as searching through the system easy to perform. Extensions of the Magic Cap interface, designed initially for inexpensive hand-held communicators rather than desktop computers, will remove many of the limitations of the current system such as the two-dimensionality of the interface, extremely small screen size and the lack of color.

In the scenario of Sally's research paper above, I have attempted to extend the direct metaphor interface pointed to by the work of the General Magic team in ways that apply

specifically to scholarly research and investigation. Thus, instead of lists of book titles, Sally is presented with piles of virtual books, grouped on tables and arranged as she has left them. Annotation is accomplished through the use of colored highlighter pens and by recording video and audio notes and sticking them to the page. Note cards are organized in groups by placing a rubber band around them. Because Sally learned these interaction patterns in the real world, she can apply them in the virtual world with confidence in the results, allowing her to concentrate on writing her paper, rather than on wrestling with the computer. Even complex concepts such as dealing with multiple sources of input (hers and her professor's) are made easy through subtle uses of color used consistently throughout the interface. Because her interaction with the computer takes place in a context that is familiar to her, Sally is able to put the system to work in quite sophisticated ways with only a small amount of cognitive load, freeing her to concentrate on the task at hand - writing her paper.

We do not need to wait for the advent of computers that are able to render photo realistic 3-D scenes in real-time in order to build such interfaces. For example, using object-oriented programming techniques, each individual element in the virtual library might know how to render itself in a variety of forms. A book may be drawn in four different views: from the front where one would see the cover graphic, from the spine for vertical placement in a shelf, in an isometric view as when piled on a table-top and opened up as for reading. The view that the user sees at any given time is dictated by the interaction they will have with the book - searching through a shelf, rearranging piles of books on a table, or actually reading the book. These four basic forms can easily be scaled and cropped to present a realistic portrayal of the book in the electronic library at the cost of little computational power.

II. Context-Based Polymorphism of Information

The second concept necessary for a complete research-to-publication writing system is context-based polymorphism of information, or allowing information to change form with time and use. In the real world, the presentation of information is often almost as important as the information itself. A fashion catalog looks nothing like a novel, which in turn looks nothing like a letter from a friend. Even though they bear a gross physical similarity - all are collections of words and pictures printed on paper - the information contained within each is quite different and is naturally presented differently. We approach these various sources of information in different ways as well, determined in part by their physical forms. One may flip through the catalog or skim the novel, but one would likely read the letter from front to back two or three times before composing a reply.

Virtually all word processing programs present data in a single, static format - a linear stream of text on a white background linked to a vertical scrollbar. Because all types of text documents are presented in the same way, word processors thus lose most of the associational clues inherent in real world information sources. Although some composition systems, notably new electronic mail programs, allow one to attach "virtual letterheads" to documents, the presentation is still essentially the same: an endless column of text on a white background.

The form factor of real-world information sources also gives clues about the information and its intended uses. Newspapers are printed on large paper with many articles per page, rendered in fairly small type to facilitate "grazing." Textbooks are smaller in size, tend to be densely packed with information, and are sometimes formatted with large areas of white space for the student to take margin notes. Magazines straddle the line between the two, densely presenting information, but in a style that facilitates both reading extended passages as well as flipping through in the quest for interesting subjects. We need to add similar metaphors to our electronic documents to help people make sense of the information contained within. This can easily be accomplished by framing the information in a real-world context. Thus in the story Sally deals with "books" and "magazines" - rather than "files"- which are consistently rendered throughout the interface. Similarly, she views each document as a series of pages rather than a scrolling column of text. This allows her to user spatial clues to help remember information, much as we do with physical books and magazines.

While applying a single metaphor is fine for static presentation of information, we must also allow the information to change form with time and use. Compiling ideas from a large set of works into a research paper is a complex

activity which demands that we go beyond the use of even static metaphors. For example, it is natural for Sally to read her source texts as "books" and make notes in virtual "notebooks." However, once she is finished with the research phase of the paper, the information begs another form. She wants to essentially stop dealing with the books and organize a loose set of ideas into what will eventually become a linear narrative. This is accomplished in the scenario above through the use of information polymorphism. She changes her notebooks into cork boards full of index cards which lend themselves to being resequenced and organized. After organizing these index cards the information is again transformed, this time into outline pages which are then be merged together into an outline for the entire paper. Finally, as she actually writes the paper, the information changes again, this time into the form of a book. The information is presented in whichever form best facilitates Sally's interaction at that stage of the research process.

It is also important to note that this information polymorphism is not a one-way process. Sally must be able to change the information back into any of its previous forms at will. Thus when the professor recommends that Sally make sweeping changes to an entire section of the paper, it is natural that she would go back to either an outline or even note card view of her data, as these forms are better suited to quick manipulation than a bound book. All tagging and organizing information must survive these phase changes so that she does not have to redo work she has already done. After "publication" these different forms may still have their uses. For example, the outline view becomes Sally's table of contents which both enables the reader to quickly get a sense of the structure of Sally's arguments as well as to quickly navigate to interesting sections of the document.

III. A Time-Based Memory of User Interaction with the Printed Word

The third concept, a time-based memory of user interaction with the printed word, is necessary for the researcher to take full advantage of an all-encompassing electronic library system. By remembering what they have read and when, we build an external, electronic memory to supplement the researcher's internal memories. This enables us to make time- as well as context-based queries of the computer. For example, Sally might say the following to her computer: "I was reading something about whales last week, on Tuesday or Wednesday, I think it was in a magazine..." This simple sentence would be enough for a publishing system to retrieve the exact passage that Sally is after. The system has a record of what she read on Tuesday and Wednesday of last week and it can narrow down an infeasible search of the entire library space to a manageable subset of information where it makes sense to look for as broad a topic as "whales."

The research process is essentially a funneling of a large set of materials through the thoughts and experience of a researcher with the goal of synthesizing these various ideas into a new whole. Our thoughts are a combination of original ideas as well as ideas from works that we have read. Thus during active research, one of the most important sets of data for a researcher is simply the set of works that they have read recently. Allowing the researcher to go back to these sources and re-trace previous paths easily will greatly increase the efficiency of an electronic library system in a very natural way. For example, keeping track of exactly how, when, and in what context searches were made facilitates Sally's later interaction with the results of the search. She might use this information to narrow down or expand the search parameters, or it may just help to remind her why she is looking at a particular set of books in the first place. Sally can also follow the chain of sources she read when she originally came up with a particular concept, reminding her where a particular insight came from and facilitating a later search for related sources.

Statistical analysis of searches and cross references also represent valuable information for the library institution. For example, librarians can use such statistics to expand their collections in areas where users show a great deal of interest or to fill gaps in the library's information space as indicated by queries that fail to bring back information. One could also envision such statistics underlying the basis of a payment system for an electronic library; compensation for an author's work could be directly proportional to the number of users who have read the work. However, these and other uses of this stored information must be balanced against the need for privacy on an individual level - issues which are beyond the scope of this paper.

While the storage requirements of an extensive map of the user's reading patterns

would seem quite large, the application of a simple 'staleness factor' to this memory data will dramatically reduce storage costs at little loss of information. For the week after a search is performed, it may make sense to cache the exact page locations of the resulting search hits so the relevant passages can be brought back into memory instantly. A week to a month after the search was initially performed, this list may be pruned back to just the list of books in which hits exist to save memory; the page and passage numbers can be quickly derived as necessary. More than a month after a search was initially run, there is very little chance that the user will need to get at the search results quickly; it is probably sufficient to simply remember the search parameters themselves and then re-run the search on demand. Finally, once the project has been completed and put away, the cached search data and parameters can be further compressed to the point of elimination to save storage space. This time-based decomposition of information, modeled after our own memory patterns, can thus be implemented at very low long-term storage cost and has the possibility of greatly facilitating the research process.

Part 3. Conclusion

The computer truly is a new informational medium, and I expect that it will change our society as much as, if not more than, the popularization of each new medium has in the past. However, I truly feel that instead of treating computers as nothing more than efficient search and retrieval engines and hanging billions of documents out in hyperspace, we should base user interaction with electronic libraries on our current research patterns. We can best do this by finding weaknesses in the paper medium that can be addressed through the use of the computer, as well as by facilitating tedious or difficult parts of the research process through the computer's aid. It is not enough to focus solely on the technological side of the issues. We must always keep in mind that humans will be inhabiting these virtual libraries and therefore that these systems should be tailored to the researcher, rather than the researcher having to adapt to needlessly complex and arbitrary systems. By remembering what the researcher has read and allowing them to retrace their steps, we supplement internal memory and thus facilitate the flow of information. By allowing the different parts of the research process to dictate the changing face of information, we give the researcher powerful tools that allow them to organize their thoughts in comfortable, natural ways. Finally, by using direct metaphors to model user interaction in the electronic library, we allow users to pull their real-world experience to the other side of the screen and free them from having to wrestle with the computer to get the job done. The combination of these ideas will empower individuals, and allow them to express themselves in this new electronic medium with grace and ease.

Bibliography

Although not directly referenced, ideas in the following list of books contributed greatly to this paper:

Brin, David. *Earth.* Bantam Books, 1988.

Laurel, Brenda, ed. *The Art of Human Computer Interface Design.* Addison-Wesley, 1990.

Moran, Daniel Keys. *The Long Run.* Bantam Books, 1989.

Moran, Daniel Keys. *The Last Dancer.* Bantam Books, 1993.

Norman, Donald A. *The Psychology of Everyday Things.* Doubleday, 1988.

Norman, Donald A. *Things That Make Us Smart.* Addison-Wesley, 1993.

Stephenson, Neil. *Snow Crash.* Bantam Books, 1993.

Sterling, Bruce. *Islands in the Net.* Bantam Books, 1989.

Sterling, Bruce. Heavy Weather. Bantam Books, 1989.

Tufte, Edward R. *Envisioning Information.* Graphic Press, 1990.

The Internet and the Aspiring Games Programmer

Ian Parberry*
Department of Computer Sciences
University of North Texas

Abstract

The Internet is an important tool for aspiring computer game programmers, providing access to information, advice from peers, and electronic publishing. We examine employment prospects in the computer game industry, resources available on the Internet, electronic publishing modes, and computer games at the University of North Texas.

1 Introduction

The computer games industry, although still in its infancy, is one of the major growth areas in computing. Current generation computer games use stunning graphics, high-fidelity stereo sound, and sophisticated scenarios. Until recently, however, the programming has been of low quality. There is an expanding market for qualified games programmers, but very little opportunity for new programmers to learn the trade.

The Internet is probably the most important tool for a novice game programmer by providing access to a large:

- repository of information about games programming tools and techniques,
- peer group of established and aspiring games programmers,
- community of end-users via shareware and freeware.

The purpose of this paper is to document the use of the Internet for computer games by the aspiring games programmer. It is divided into five main sections, covering respectively employment prospects, game information available on the Internet, electronic publishing modes, and computer games at the University of North Texas.

*Author's address: Department of Computer Sciences, University of North Texas, P.O. Box 13886, Denton, TX 76203–3886, U.S.A. Electronic mail: `ian@ponder.csci.unt.edu`. URL: `http://hercule.csci.unt.edu/ ian`.

2 Employment

Since the computer games industry is in its infancy, games companies looking to hire programmers are interested in experience rather than college degrees. Submitting a polished resume is less important than submitting a disk of computer games that the applicant has written or collaborated on. But this situation will probably change as the industry matures. It is plausible to expect that within a decade, game companies will be looking for employees who have produced great games *and* have college degrees. The type of skills that a prospective games programmer needs that can be gained at a university include:

C/C++ Programming: For portability, games should be programmed in a high-level language. C and C++ seem to be the most popular.

Assembly Level Programming: For speed, the low-level aspects of high-performance games have to be programmed in assembly code.

Computer Architecture: Games programmers need to take advantage of advanced hardware features, for example, clocking, caching, DMA, interrupts, and RISC.

Software Engineering: Games programmers seldom work alone. This can cause major problems for programmers who are not experienced in producing commercial quality software to a deadline. Modern software engineering techniques can address some of these problems.

Computer Graphics: Stunning computer graphics and animation are a major factor in selling a game. A syllabus containing elementary 2d material plus advanced 3d material including shading and rendering is most useful.

Algorithms and Data Structures: A knowledge of standard algorithmic techniques and data structures will save the game programmer from having to constantly reinvent the wheel.

Communication Networks: This class is a must for multiplayer networked games, which are becoming

among the most popular.

The additional game-specific material that games programmers need can be obtained from one of the recent spate of books on the subject, such as Gruber [2], Hook [3], Howard [4], LaMothe [5, 6], Lampton [7, 8], and Robert [12]. The Internet also provides many important resources, which are discussed in the next section.

3 Game Information on the Net

3.1 Newsgroups

There are more than a hundred newsgroups that are relevant to games, mainly in the `alt.games.*`, `rec.games.*`, and `comp.sys.*.games` hierarchies. Also useful are the newsgroups in the `comp.graphics` hierarchy for computer graphics techniques. The most important newsgroup for the games programmer is `rec.games.programmer`. However, the traffic in that newsgroup is high, and much bogus and misleading information is posted there. Source code for some games are posted to `comp.sources.games`.

3.2 `ftp` Sites

Much information about computer games is available by anonymous `ftp`. Table 1 lists a few common `ftp` sites, the best of which is `x2ftp.oulu.fi`. Be warned however that some of the material out there is pirated, copyrighted, or just plain illegal. Games programmers should not include code, images, or sounds taken off the net in their game unless they are certain of its provenance. Big companies can and do sue to protect their rights to the material regardless of the monetary cost or expected return.

For the games programmer, the biggest resource is the PC Games Programmers Encyclopedia [1] which is a collection of text files of varying quality written by many different authors covering various aspects of programming games for the PC.

3.3 WWW Sites

Some game information is available on the World-Wide Web. Table 2 gives some useful URLs.

4 Publishing Modes

Electronic publishing of computer games typically utilizes bulletin boards, Internet `ftp` sites, and commercial online services (such as Compuserve). Several different modes of publishing have emerged:

Freeware

A *freeware* program is one that is distributed electronically with no payment required. This avenue can be used by a beginning games programmer who is building a portfolio of games that can be used to impress a prospective employer. The term "freeware" has been trademarked by Andrew Fluegelman, an early shareware pioneer.

Shareware

In contrast with freeware, a *shareware* program includes a legalistic or pseudo-legalistic request or demand for payment. The user who sends in a payment is said to have *registered* their copy of the game. Game authors can actually make a living doing this, but registration rates are typically very low. Registration is encouraged by offering upgrades or full printed documentation (see also nagware, crippleware, and heroinware below).

The shareware user is encouraged to share the unregistered version of the game with friends, thus building up a loyal customer base without the expensive overhead of advertising, packaging, or negotiating legal contracts with distributors. The downside is that many shareware authors report that registration rates are low. Reported figures range from 1% to 80%, but there is typically little evidence on which to base these conjectures.

Nagware

Nagware is a version of shareware that encourages registration by popping up a nag screen that reminds the user to register when they first start up the game. Typical approaches include locking the computer for a small period of time (typically 5 to 30 seconds), or by hiding the button or key-sequence that the user needs to make the nag screen go away, with the intention of making the user read or scan the whole screen in order to find it. The downside is that nag screens are either so innocuous that users ignore them, or so obnoxious that users remove the game from their hard drives. The middle ground is very hard to find.

Crippleware

Crippleware is a version of shareware in which the program is crippled in some way, usually by disabling play features, or by preventing the user from saving a game. The consumer must register the game in

`cs.columbia.edu`	game FAQs
`faui43.informatik.uni-erlangen.de`	Unix games
`ftp.cica.indiana.edu`	microsoft windows games
`ftp.uml.edu`	DOS games
`ftp.uwp.edu`	id archives; some development tools
`infant2.sphs.indiana.edu`	doom
`nic.funet.fi`	Amiga, DOS and Unix games
`rtfm.mit.edu`	game FAQs
`sumex-aim.stanford.edu`	Macintosh games
`wuarchive.wustl.edu`	dos, general
`x2ftp.oulu.fi`	the best general site

Table 1: Some game `ftp` sites.

`http://wcl-rs.bham.ac.uk/GamesDomain`	the best place to start
`http://www.cm.cf.ac.uk:/Fun/`	pointers to games (esp. unix)
`http://www.fokus.gmd.de/minos/employees/hgs/audio/audio.html`	PC audio hardware
`http://obsidian.math.arizona.edu:8080/netrek.html`	Netrek, net-based Star Trek
`http://hercule.csci.unt.edu/larc`	UNT LARC

Table 2: Game-related URLs.

order to get a fully functional copy. The consensus of opinion among game players is that crippleware is so annoying that they never play more than once, and never register the copies.

Heroinware

Heroinware is a new variant of shareware pioneered by *id Software* with their game *Doom*. Unlike crippleware, the executable code for the entire unaltered game is distributed electronically, with enough levels to allow the user to play for an extended period of time. The user gets the remaining levels by registering the game. The rationale is that users play for long enough to get hooked on the game, and purchase the remaining play levels. Where crippleware is designed to prevent the user from enjoying the game without paying for it, heroinware encourages the user to enjoy the free version, in the hope that he or she will want more. *id Software* have reported that 127,000 registered copies of *Doom* have been sold, and that approximately 10 million unregistered copies exist.

In yet another marketing coup, *id Software* have released *Doom 2* as a full retail program without using shareware. An estimated 600,000 copies were ordered before the game was released, generating over \$41 million in revenue at the \$69 per copy list price. It may be hypothesized that this incredible success (in the games industry, 200,000 copies is considered a blockbuster) is due to the large customer base built up through the use of heroinware.

5 Computer Games at UNT

5.1 The LARC Project

The author of this paper established the Laboratory for Recreational Computing at UNT in 1993. Membership in LARC is open to undergraduate and graduate students at UNT, and is on a voluntary basis. Current membership is almost exclusively undergraduate, and includes computer scientists, artists, and musicians. The group meets formally for two hours once a week. UNT has provided laboratory space and five computers with sound cards and joysticks. LARC members have keys to the laboratory and have exclusive access to the equipment at all times. Membership figures are shown in Figure 1. More information is available on LARC on the World-Wide Web [10].

5.2 A Computer Games Class

A senior level undergraduate course on Game Design and Programming was offered at UNT in Fall 1994 under an experimental `CSCI 4980` course code. Preliminary announcements on the Internet in `unt.general`, `rec.games.programmer`, and

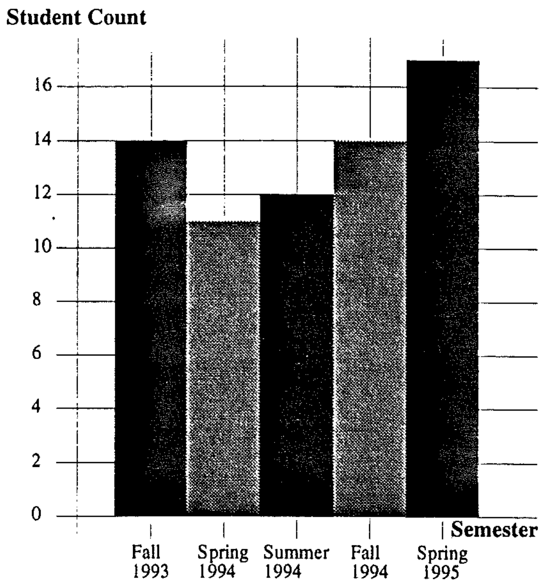


Figure 1: Student membership of LARC. Figures for Spring 1995 are anticipated from a poll of students.

`dfw.general` attracted 90 queries for information, and 24 students were enrolled, (8 of them already LARC members). The course was taught without a textbook, relying heavily on resources found on the Internet. More information about `CSCI 4980` is available on the World-Wide Web [9]. It will be offered again in Fall 1995 under the `CSCI 4330` course code.

Material covered in the course fell into two categories, nontechnical and technical. The nontechnical material included:

- game genres
- marketing and copyright
- introduction to the internet
- choosing a name
- sex, violence, and political correctness
- what makes a successful game
- game reviews
- storyboarding
- the game proposal
- team composition

The technical material included:

- introduction to 32-bit programming
- graphics file formats
- offscreen buffers and blitters
- sprite animation
- palette manipulation
- 2d graphics
- input devices: keyboard, mouse, and joystick
- 32-bit protected mode assembly programming
- the physics of sound sampling
- sound effects using VOC files and DMA
- general MIDI
- 3d graphics techniques
- code optimization
- hooking to timers
- mode X graphics

Grading was based on two simple programming assignments, and the major team project. The first program was to add features to a simple sprite-based game engine, and the second was to add sound effects to the same program. The team project required the design and implementation of a prototype for a new computer game. Students worked in teams of one to three members to submit a proposal, a storyboard, a progress report, and the final program.

5.3 Feedback

LARC has been in operation for long enough that feedback from students, administrators, and faculty can be evaluated. Each of these will be discussed in a separate subsection. Faculty planning to duplicate LARC should be aware that although students will be in favor of it, faculty and administrators may be unwilling support it.

5.3.1 Student Feedback

Student reaction to LARC has been exclusively positive, even from students who have no interest in computer games, and those who have an interest in computer games but choose not to participate. Student reaction to the `CSCI 4980` class offered in Fall 94 was enthusiastic, including statement such as the following (an unedited transcript of anonymous student comments is available on the World-Wide Web [11].)

"The Entertainment Industry has long been a closed market to those who follow the educational route career, one this class succeeds in opening up these doors, and giving insight into one of the fastest growing software markets."

"This was an excellent and informative class."

"This class (and the chance it provides) are, without a doubt, absolutely necessary."

"This is one of the most interesting and practical classes that I have ever taken."

"I have already learned more in this class than I would have expected to learn in any class I had taken."

"This is by far the most informative computer science course (or any other type of course) that I have taken anywhere."

"Actually, though I had researched this topic for some time before taking this class, my eyes were opened to the world of techniques, tools, and methods used to create software in this field (games programming). This class has armed me with information extremely well, and pointed out where to look for more."

5.3.2 Reaction from Administrators

Reaction from administrators at UNT, from department chairs to Deans to the Chancellor, has also been very positive and encouraging. The Computer Science department has been generous in providing laboratory space and funds for an initial purchase of computer equipment. Higher administrators have been supportive of the idea of using LARC for promotional purposes, which benefits LARC directly by attracting more students.

5.3.3 Reaction from Faculty

Reaction from faculty has been more guarded. Curiously, faculty outside the Computer Sciences Department have been more supportive of LARC than those inside it. The major criticism from faculty in Computer Sciences is that games are not appropriate to the Computer Science curriculum, which as we argued in Section 2, is not true. Games programming is an expanding part of the employment market, and can be integrated into a traditional Computer Science curriculum as a capstone course that integrates knowledge gained at all levels.

References

[1] The PC Games Programmers Encyclopedia. Version 1.0, available by anonymous `ftp` from `teeri.oulu.fi` in `/pub/msdos/programming/gpe`, 1994.

[2] D. Gruber. *Action Arcade Adventure Set.* Coriolis Group Books, 1994.

[3] B. Hook. *C++ Game Programmer's Secrets: What the Game Companies are Afraid You'll Find Out.* Sams Publishing, 1994.

[4] C. Howard. *Programming Games for Beginners: Visual Basic for Windows for Fun and for Profit.* Sams Publishing, 1993.

[5] A. LaMothe. *Teach Yourself Games Programming in 21 Days.* Sams Publishing, 1994.

[6] A. LaMothe, J. Ratcliff, M. Seminatore, and D. Tyler. *Tricks of the Game-Programming Gurus.* Sams Publishing, 1994.

[7] C. Lampton. *Flights of Fantasy: Programming 3-D Video Games in C++.* The Waite Group Press, 1993.

[8] C. Lampton. *Gardens of Imagination: Programming 3D Maze Games in C/C++.* The Waite Group Press, 1994.

[9] I. Parberry. CSCI 4980: Computer Game Design and Programming. A WWW document with URL `http://hercule.csci.unt.edu/larc/4980.94f`, 1994.

[10] I. Parberry. The Laboratory for Recreational Computing. A WWW document with URL `http://hercule.csci.unt.edu/larc`, 1994.

[11] I. Parberry. Student Comments from CSCI 4980, Fall 1994. A WWW document with URL `http://hercule.csci.unt.edu/larc/4980.94f/feedback.html`, 1994.

[12] D. Roberts. *Easy PC Game Programming.* Coriolis Group Books, 1994.

Digital Libraries and Large Text Documents on the World Wide Web

Harry Plantinga
University of Pittsburgh
Department of Computer Science
planting@cs.pitt.edu
http://www.cs.pitt.edu/~planting/

Abstract

The World Wide Web (WWW) has strengths and weaknesses as a delivery vehicle for digital libraries. This paper discusses experiences with a small digital library on the WWW and describes some of the problems encountered. One problem in particular is addressed: that of the HTTP data delivery model, in which entire documents are transferred and displayed. This model is not ideal for large reference documents such as encyclopedias, dictionaries, and commentaries. This paper describes the approach taken to address this problem, of paging large documents into smaller HTML documents, while ensuring the validity of the returned HTML sub-document and minimizing the load on the server.

1. Introduction

The Christian Classics Ethereal Library (CCEL) is a small, experimental digital library on the World Wide Web (WWW) [1]. Its purpose is in part to experiment with electronic publishing and digital libraries on the WWW. It was started in May 1994, and as of February 1995 it had 26 HTML books and hundreds of other books and documents in text, HTML, RTF, PDF, and other formats. The access rate has been increasing by about 50% per month of late, reaching 70,000 for February 1995.

When people hear about the existence of a library on the WWW, they often make a comment along the lines of "Ugh, who would want to read a book on a computer screen?" I have sympathy for that point of view. The longest stretch of reading of a single book on a computer or PDA screen that I have managed is about an hour, on an Apple Newton PDA. But that is not the way this library is most commonly used. The most common use is accessing reference works, where only a small portion of text is needed and the searching and indexing capabilities of computers are most useful. These reference works include traditional reference books, such as a dictionary or commentary, and new reference works which contain links to other items on the Internet. Another common use is for browsing books, which can be downloaded and printed if they are of interest. Also, on-line

libraries serve as the destination of hypertext links from other works.

The construction of this library has made evident many of the advantages and disadvantages of the WWW as a vehicle for digital libraries. Some of its advantages are that HTML represents the first widely-used open standard for text mark-up -- previously most documents widely exchanged on the Internet were ASCII text. Hypertext links are useful in digital libraries for footnotes, hypertext tables of contents, and references to external documents. High quality, freely available web browsers are available. The forms interface and CGI programs make it possible to do things that books have never before done.

One factor that many editors will consider a disadvantage is that they will have to give up much control over the appearance of the books they prepare, since HTML is essentially a structural markup language. Another disadvantage is that URLs specify locations on the Internet. It would be convenient to be able to have hypertext links that refer to a particular document rather than a particular location: local or regional copies, mirror sites, and backup servers would be useful for large documents located on a single server which may be heavily used. To address these needs, universal document identifiers such as the Universal Resource Numbers of HTML+ would be very helpful.

The transmission vehicle for the WWW is the Internet, and most people who have access to the Internet have it at the office or university in the form of a fast connection to a local area network which is connected to the Internet. This is a disadvantage in that books that are read via the WWW are usually read on the screens of desktop computers. Probably because this is unaesthetic if not injurious for long periods of reading, books are often just browsed or read for a short while. Of course, the transmission vehicle for the WWW may become a great advantage if wireless connections to the Internet become common. Then a small handheld bookreading device such as an Apple Newton would be almost as usable for bookreading as a traditional book. Although traditional books will still have advantages such as the sharpness and contrast of the page, a bookreader on a wireless network will have other significant advantages, such as small physical size, hypertext links, and the free and instantaneous availability of thousands of classic books and reference works.

2. Large Documents on the WWW

However, the problem that I wish principally to address is that the hypertext model used by the WWW is not ideal for digital libraries. The model used is that of transmitting and displaying an entire document when a link is activated. Some browsers improve the model by displaying a partial document while it is being downloaded in the background. But even in that case, if a particular location in a document is referenced by a hypertext link, nothing can be displayed until the entire document to that point has been transmitted. This is inconvenient for large documents such as books, where it may take minutes to download the document before it is available for reading. But it is especially inconvenient for large reference works, which may constitute many megabytes of data, and of which only a very small section may be of immediate interest.

The method of dealing with this problem suggested in HTML writing guides [2] is to break up large documents into a number of smaller documents. But this approach degenerates to impracticality for large reference works. Imagine, for example, a dictionary with 20 M bytes of data and 100,000 entries. Should it be divided into 100,000 separate files? The allocated but unused disk space alone would be 1,580 M bytes on a file system with 16 K byte allocation units. It may also be that the file system of the server

will prove inefficient at indexing and accessing such a large number of files. In addition, creation and maintenance of such a large number of files would be difficult and slow. Should the dictionary be stored with a smaller number of larger files? Much data will needlessly be transmitted across the Internet and users will have to wait longer than necessary for desired information to appear.

Then too, users often want to print out a large document such as a book, or perhaps an extended section of the document. If it is stored as a web of files, printing is problematic. One could maintain separate versions of the document, one for printing and one for the web, but that also has drawbacks.

The approach taken to address this problem in the CCEL is to store large reference works as one or a few large HTML files. A CGI program [3] is then used to select the desired section of the document and return it only as an HTML document. I call the program that selects and returns small pieces of a large HTML document the "pager".

2.1 Subdocument Addressing

In order to return a subdocument it is first necessary to be able to specify the text extent of interest. The HTML 2.0 DTD draft [4] offers a name attribute for an anchor as a means of naming a section of text. For example,

```
<A NAME="Section1">This is
section 1.</A>
```

Unfortunately, the text between the `<A>` and `</A>` tags is not supposed to contain anchors, and nested anchors give unpredictable results on some browsers. Therefore, sections containing anchors cannot be surrounded by `<A> ... </A>` tags. In practice, named anchors are used to signify locations in a file rather than text extents. Thus, text extents must be specified with a beginning point and an ending point.

The HTML 3.0 DTD draft of 19-Jan-95 [5] changes the status of the `NAME` attribute for anchors to "deprecated." In its place, an `ID` attributed is supported for most elements. The `ID` attribute can be used in place of `NAME` to mark the require points in the text.

2.2 Returning a Section of a Large Document

The other need identified above is the ability to return a small section of an HTML document. However, just returning a specified section leads to a couple of problems. First, a section of a valid document HTML may not be valid HTML. For example, a tool for converting files in a word processor format (RTF) to HTML might convert a heading in the word processing file to HTML like this:

```
<h1>
<a name="RTFToC4">2.2
Returning a Section of a Large
Document
</a></h1>
```

Suppose the anchor were used as the starting point of a section of the document. The start of the section returned to the user would be

```
<a name="RTFToC4">2.2
Returning a Section of a Large
Document
</a></h2>
```

Notice that the `<h1>` start tag is missing, while the `</h1>` end tag is present. The returned section is not valid HTML, and furthermore it would result in wrong rendering by most browsers. In general, a section of a file returned in this manner would also lack the `<HEAD>` section of the document and any HTML container tags that were not terminated by the start of the section. Therefore it is in general necessary to parse the original file and return any required start tags, the selected extent of text, and any required

end tags. It may also be desirable to prepend a header and other information to the text.

3. A Pager for Large HTML Documents

The design goal for the pager was that it return pages from documents stored as standard HTML rather than some other format. Therefore the HTML named anchor facility was used for identifying the beginning and end of sections of text. In essence, the idea behind the pager is simply that it returns the section of an HTML document between a `<A NAME = "section_name">` tag and the next `<A NAME = "">` tag or the section between two specific named anchor tags, along with some header information.

Since the part of the document between these tags may not be valid HTML, the pager could parse the document to add any required tags, or a preprocessor could be used to modify the file so that sections between named anchors represent valid HTML. However, not all of this work may be necessary in practice if the HTML document is constructed in such a way that the named sections constitute valid HTML.

Another design problem to address concerns efficiency. It is clearly undesirable to return an entire encyclopedia when one article is requested, but it is also undesirable for a pager program running on a server to read sequentially through an entire encyclopedia to find a requested article. This problem may be alleviated by breaking up a large document into a few smaller documents. A better solution is to have the pager program automatically construct an index to a large HTML document the first time it is read and parsed, storing the starting and ending character positions and the tags to be prepended and appended for each named section. Later accesses are achieved by reading the index file and directly returning the part of the document requested without parsing. If file modification is detected, the index is rebuilt.

Many additional features that are useful for bookreading can be added onto this basic structure. Forward, backward, beginning, and index buttons would obviously be useful. Other features might include adjustable parameters for characteristics of performance such the preferred number of pages to return at one time, whether to include footnote texts at the bottom of each page, whether to display a progress meter, and so on. Context information, such as the current page and user preferences, may be specified in the HTTP query:

```
pager.cgi?file=book.html&from=sec
tion_1&to=section_3&footnotes=fal
se&meter=true
```

The prototype pager [6] returns the document head and the part of the body before the first named anchor. It does not parse or construct indexes, and the only additional features it currently provides are forward, backward, beginning, all, and up buttons and the ability to specify *from* and *to* section names. Nevertheless, it makes the use of digital books on the WWW much more efficient, and users say they love it. Formatting a book for use by the pager only involves inserting named anchors to delimit pages in such a way that paired start and end tags don't span them.

4. Conclusion

The pager has made the use of the WWW for a digital library much more practical. Even slow Internet links are suitable for bookreading when small sections of a book can be accessed individually. Some of the particular benefits of the use of the pager are that large documents can be stored in one or a few files; small sections of a large document can be referenced and retrieved individually; it is not necessary to transmit an entire book up to a point in order to start reading it at that point; and HTML links can be made to locations *inside* a book without

the concern that following the link will download an entire book.

The pager is especially useful for large reference documents such as dictionaries, where small sections of the text are desired. It is possible to construct a complete index file in such a way that following a link downloads only the article of interest. Page forward and backward buttons can be used to browse the dictionary. If the index file is large, it too may be paged, resulting in a two-level index.

A disadvantage of using the pager is that named anchors must be inserted into the HTML document to delimit pages and an additional program must be run on the server for every page returned. Users are not eager to add named anchors to books by hand; a utility for normalizing and adding anchors would be a useful addition.

It is currently unaesthetic to read books on most computer screens in most cases. However, the situation is likely to change dramatically as hand-held computers suitable for bookreading become more common and more commonly connected to the Internet. Then it is likely that the large number of books and reference documents already available on the WWW will make its use for bookreading and digital library access grow even more dramatically.

I conclude with a plea to browser authors: support the HTML link tags for previous, next, and parent documents, e.g. `<LINK HREF="docname" REL="next">`. Support the navigation to those documents with left, right, and up arrow keys, and scrolling down and accessing the next document by pressing the space bar. Finally, offer an option of automatically pre-fetching the next page. Then nearly all network delays and mouse actions for remote bookreading would effectively be eliminated.

Notes

[1] It is available at `http://www.cs.pitt.edu/~planting/books/`

[2] CERN's HTML+ documentation, for example, states that "Keeping a large document such as a book in one node will increase the time it takes to retrieve the node over the network. It is generally better to split large documents into a number of smaller nodes" (`http://info.cern.ch/hypertext/WWW/MarkUp/HTMLPlus/htmlplus_9.html`).

[3] CGI programs are programs that can be written for the world wide web that in response to query strings generate and return documents on the fly.

[4] The March 7, 1995 draft, available from `http://info.cern.ch/hypertext/WWW/WWW/MarkUp/html-spec/html.dtd`

[5] Available from `http://info.cern.ch/hypertext/WWW/MarkUp/html3-dtd.txt`

[6] Available from `ftp://kuyper.cs.pitt.edu/ebooks/HTML/pager.cgi`

Making Multimedia Work For Women

Adrienne GreenHeart
College of Liberal Arts, Boston University

Most interactive software aimed at entertaining is aimed at men. The current interactive entertainment industry—primarily games—is evidence of how quickly women can be excluded from a new technology if women do not take part in its development. Brenda Laurel describes how the game industry evolved into an industry that caters to men:

> In 1976 I got involved in the computer game business. I learned from my bitter experiences there that what you do with a medium early-on, and who gets access to shaping it, has a huge effect on the kind of message and experiences that the medium is capable of supporting.

In the beginning of computer games (the cartridge and floppy market), men worked on the software, so the industry became a vehicle for male expression. In order to avoid the recurrence of this situation, both men and women should insure that women become involved in the development of new media (the CD and broadband market).

This paper presents some ways that multimedia is an appropriate venue for feminine story-telling; not stories that focus on games, but stories that focus on the intent of the artist to express an idea or vision.

Major questions to ask, in terms of developing multimedia as an art form are, What are the unique qualities of multimedia? And what are these qualities good for? Two unique qualities of multimedia are randomness and interactivity—capacity for non-linear presentation by way of the computer determining the order of presentation or the user determining the order of presentation. Finding what they are good for is more difficult.

One answer is that interactive media is well-suited to the messages and experiences of women's fiction—video or text. The issues that women deal with in their fiction, and the structure of women's fiction begs for presentation in a non-linear format. <u>The Columbia Literary History of The United States</u> describes women's fiction as creeping farther and farther away from traditional forms:

> It quickly became obvious to a number of important female authors that the basic assumptions and conventions underlying realistic fiction—its reliance on reason and causality, its central myths, its requirements for a dramatic action in which conflicts could be resolved, its implications about what constituted "heroism" and "significant" action—were inherently male-defined and hence in many ways inadequate to convey the most salient features of women's lives.

This explanation of women's writing shows that basic assumptions and conventions underlying linear fiction are not appropriate to some experiences. Many women authors have found non-linear narrative more appropriate for their fiction, (for example Kathy Acker, Jeanette Winterson and Virginia Woolf). I will present some examples where non-linear story telling suits womens' lives:

• While there is a cyclical nature to all of life, the female life is especially cyclical, and therefore often difficult to fit into linear narrative. Linear formats are metaphors for the idea that life is a straight path from one end to another. Multimedia avoids these metaphors, and provides a new opportunity for women to explore the cyclical nature of their lives.

• The linear story line also assumes linear time, which goes hand in hand with the concept of pastness. Therefore, linear time discourages women from redefining themselves outside the patriarchy, which has permeated the past. Victoria Smith writes:

> The anticipation of the present's future status as a memory gives the present moment a quality of pastness. That same anticipation posits a relationship between a series of present moments, each of which contains its own pastness. This experience of time then is a way of living historically, or inscribing one's present into history and assuring a future.

Hence, linear time places us in a constant state of pastness; as soon as a moment exists, we have moved on to the next, future moment. In the past, women have often been forced to marginality. In order for women to see themselves in a new way, women must not be made to exist in the past. So in as much as a woman defines herself through her fiction, she must create a new sense of time in order free herself from marginality. Multimedia opens up new models of time representation and therefore new contexts for women to redefine themselves.

• Women authors are also experimenting with non-linear alternatives to the male, authoritative voice. An example is Susan Griffin. In her book, Women and Nature, Griffin uses non-linear exposition, and suggests that non-linear writing shows there is not one, elevated truth, but many truths. The idea behind this exploration is that truth is dynamic, and multimedia provides a dynamic and therefore appropriate space to engage in the deconstruction of this authoritative voice.

• Other women are writing in a private, confessional vein. This writing is an example of subject matter that pours out in a jumble, not in a linear story curve. This is fiction about private journeys of the soul rather than public journeys with physical destinations. Reality in a private journey is not linear, but rather a layering and intermingling of different realities. Multimedia can present the private journey as a series of criss-crossing lines and collage-like experiences, rather than a single line of narrative.

• The private journey of many womens' writing transgresses typical language boundaries of linear narrative; traditional narrative language is a public language, not meant for personal use. The patriarchy hides behind its language, which is linear, and the dominant language never has words for the feelings of struggle against the dominant ideology. Through the use of random juxtaposition of texts or text and images, multimedia can challenge the patriarchy by finding words and images to break through the language barrier. Multimedia encourages non-linear expression, and language that transgresses the patriarchy.

• Patriarchal language also depends on a system of symbols that shape how women see themselves. For

example and advertisement that juxtaposes a woman and a car symbolizes that idea that a man can buy both: women next to cars symbolize male power. Interactive fiction facilitates the deconstruction of this symbolic language: Randomness within a context of story space creates fresh juxtapositions that challenge conventional meanings; and interactivity encourages active use of language rather than passive reception. With both the randomness feature and the interactive feature, this fiction permits a new type of union between words and pictures, thereby creating a new language.

Multimedia can bring non-linear expression into the center of the art world. And if women involve themselves now, when the art form is just beginning, the media will work for women. Women should use this new tool to better express what women are already expressing. In a world where male language dominates and stifles woman's expression, interactive fiction is a rare opportunity for women to begin to shape a new form of expression. Women can widen multimedia from a medium dominated by male games to a medium that includes art with a message—a female message.

Bibliography

1. Emory Elliott, ed. The Columbia Literary History of the United States, Columbia University Press: New York 1988, p. 1170-1171.

2. Susan Griffin, Women and Nature, Harper Row: New York 1978.

3. Laura Owen, Her Blood Is Gold, Harper-Collins: London 1993.

4. Jeanne Siegel, ed., Art Talk: The Early 80's, Da Capo Press: New York, 1988.

5. Victoria Smith, "The Text of Her Self: Be-ing in Sister Gin," unpublished essay 1985. AQI June Arnold, Sister Gin, The Feminist Press: New York 1975, in the afterward by Jane Marcus p. 239-240.

6. Wired Magazine, March 1993.

PH Model: A Persistent Approach to Versioning in Hypertext Systems

Georgia Panagopoulou, Spiros Sirmakessis, Athanasios Tsakalidis

Department of Computer Engineering and Informatics
University of Patras, 26500 Patras, Greece
and
Computer Technology Institute
PO Box 1122, 26110 Patras, Greece
e-mail: panag@cti.gr
tsak@cti.gr

Abstract

In this paper, we present a general data model for a *persistent* hypertext system. We describe a model that can handle versioning in an efficient way (efficiency is defined in terms of minimising the space requirements). An extended description of the way, insert and delete operations should be handled, is presented in order to achieve efficient space and time bounds in storage and retrieval of the whole history of a piece of information.

1. Introduction

A great amount of hypertext systems have been developed and commercial systems such as KMS [1], Intermedia [11] and Notecards [13] are now available to the common user. An abstract representation of the internal structure of a hypertext system, called the data model, lies inside every hypertext system. In this work we describe a new data model, which supports versioning with efficient space requirements.

The PH data model stands for Persistent Hypertext and provides the ability to keep the history of all the states of a hypertext system, along with other capabilities of a common hypertext system. By the term system's history we mean the various transformations of text, pictures, etc. that were saved as a system's state in the past. In order to achieve this requirement we use elements of the theory of persistence ([8]) and we conclude in the definition of a persistent hypertext data model. This means that any time, a user has the ability not only to access previous versions of the data, but also to update any of these versions. This is different from the meaning of versioning as it has been implemented in the past. Versioning is the system's ability to support the saving and retrieval of different versions of elementary pieces of information. Persistence requires the system to keep track of all the system's states, which would be available as an entity to the user for reading and modification at any time. Moreover it supports versioning in an efficient way, meaning that we can store every version of the data at O(1) amortised time and space cost and retrieve or update any version with O(1) worst case time cost per access step.

2. Preliminaries

The model presented here is based on the theory of persistence, as it is described in [8]. According to this theory, a structure is called *persistent* if its history is not lost, as the structure is modified through time. Persistence can be divided into two different forms:

- partial persistence
- full persistence

We call a structure *partially persistent* if all of its versions are accessible but only the last one can be updated.

On the other hand, *full persistence* refers to structures having all versions available to any future operation. This means that update operations can be performed on any version of the structure.

The work of Driscoll, Sarnak, Sleator and Tarjan in [8] gives an optimal solution to the problem of keeping the history of a structure with minimal space cost.

Current hypertext systems have very limited version control. There are non-versioning systems, such as Notecards ([13], [14]) and gIBIS ([3]), that do not specify the history of the system. KMS ([1]), provide annotation features, as a substitute for versioning, which is not an efficient solution.

Among the systems that support versioning, the only system that gives serious consideration to versioning is Neptune from Tektronix ([7]). Neptune supports linear versions for nodes and links. It also provides the ability to point at the current version of a node. But this is different from the "full" versioning mechanism that we would like to provide. Another system that is quite interesting is the PIE system from Xerox. Although strictly speaking, it is not a general hypertext system, but a program development environment for SmallTalk, PIE is nevertheless widely referenced in hypertext literature because some of its concepts are relevant to hypertext systems in general. It supports layers grouped into contexts. When a context is modified, PIE creates a new layer. We do not consider this approach as versioning control.

The problem of providing different views of the hypertext network was also addressed in the design of the Intermedia system ([11]). Intermedia allows the user to collect all the links defining a particular view in a *web*. Webs are sets of links that connect the various nodes, that are accessed by the user. We can thus construct versions of the network topology by using different webs. This, along with a linear versioning of the nodes themselves, gives the user a basic and simple versioning capability.

The HyperPATH/O2 system ([2]) contains persistence features, but handles persistence in a completely different way without versioning support. Interesting models, such as the HM data model ([17]), HAM ([4]) and the model presented in [15], do not give any clue for versioning.

The "New hypermedia data model", presented in [15], mentions persistence as a basic aim of its design but it gives a different meaning of persistence instead of the original one presented in [8].

All the systems described above, support access to older versions of the system, but only the modification of the most recent is available. Our model supports versioning in a way that the modification of any version of the system is also available.

This paper is organised as follows. Section 3 gives a short description of the PH data model. Section 4 contains examples of the way queries can be handled and we point out the differences among PH model and other models. Next, in section 5 we propose areas where the use of the proposed model could make a hypertext system more useful.

3. The Data Model

The data model is one of the most important elements of a hypertext system. It is an abstraction of the internal structure of the hypertext system and it gives a schematic representation ([19]) of the basic operations supported.

The main goal of PH model is to maintain the history of data. This means that the PH model preserves all the states of a system, from the moment it started its operation. This feature is quite significant for many applications (e.g. accessing different releases of software packages, providing backup facilities, estimating previously done work etc.). This can be seen as a simple operation of versioning, as this is used in many hypertext systems. The basic difference between the model described here and other existing hypertext systems is that the PH model is based on a fully persistent data model. Persistence guarantees efficient time and space requirements in every operation. The space required to store all versions of the model is linear (in terms of the number of the stored nodes) and the worst-case time for access to any version is O(1) per access step. Moreover update operations can be done in every version of the model.

The PH data model consists of a set of nodes and a set of links. These two sets are combined together in such a way that a non-linear graph is created. Every node of the graph contains an elementary piece of information. This information could be a single **frame**, consisting of pure text, pictures or text and pictures together.

The links are used to connect the nodes with each other. They also contain information about the content of the nodes they connect. This means that the links have some kind of internal structure. We can move from a node A to a node B through a link and vice versa, meaning that the connection between two nodes is bi-directional.

Depending on the kind of connected nodes, two different types of links are used. These are:

- **Persistent links:** they are general purpose links that are used to connect the nodes of the graph. These are the basic links of the graph and have internal structure. They carry information about the nodes. This information is separated into four different categories, which are:
  - i) authority/access permission
  - ii) aggregation
  - iii) version number
  - iv) deletion/update mark

  The first field contains information about the different access modes of a frame. It describes whether a node can be written, deleted or even read.

  The aggregation field is filled only when the connected nodes are to be used as a generalised node. If this is not the case, then the field is left empty. Due to the aggregation property, the nodes linked together as aggregated, compose a set, which is treated as a single "super" node (e.g. this is the case of a whole chapter, distributed among different nodes). This notion makes use of the hierarchical structure of hypertext systems and leads to a more strictly hierarchically organised system. The basic advantage of aggregated nodes is that they provide access to data to different users with different levels of abstraction ([16]).

  The third field denotes the number of the version, in which the node was created or a crucial change in the state of the system happened. This number is not unique because the connected nodes may appear in more than one versions of the system. More details of these notions can be found below. Version numbers are kept in memory in a way that allows quick access and search. This is described in the next section.

  Deletion/update marks are used as semaphores to denote whether a node is deleted (or updated) or not. In case the content of a node has changed, this field (on every link that leads to this node) is set and contains the number of the latest version the node was part of. The use of this field is to inform the system whether the accessed link

leads to a node contained in the current version or the node has been updated in an older one. More details are listed in the next section.

- **Version links:** these links are used to connect together different versions of the same node. More precisely, they are plain pointers with simple internal structure. They contain only the version number of the frame they are linked to.

4. Queries in the PH Data Model

In this section we explain the way the model handles insert and delete operations, giving schematic diagrams, wherever they are needed.

In Figure 1 appears a snapshot of the graph. For convenience we write on the links only the version number. In this example only one version exists, version 1.

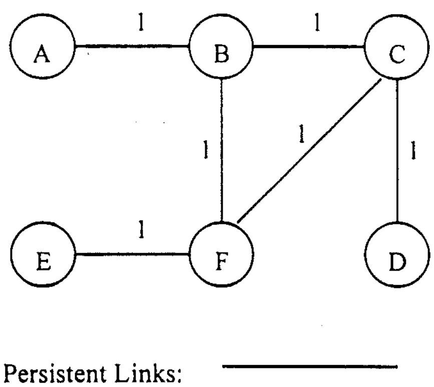


Figure 1: A simple hypertext graph

Consider now that we access node F and modify the frame stored there. In this case a new node F′ is created. This node contains the new modified data of node F. The node F′ is connected with node F (old version of the frame) through a version link. Moreover it is connected with all the nodes that node F is connected, through persistent links. These new persistent links are assigned to a new version number. If the old version was version n, then the new version number is n+1. The "update mark" field of the links used to connect node F with the rest of the graph, is now marked as updated. The field's value is the number of the version, when the update has occurred (e.g. n). The rest links of the graph are not influenced.

The new form of the graph is shown in Figure 2. The links that connect node F with B, C and E contain an asterisk, meaning that the node had been updated, and a number indicating the last version when the link was used.

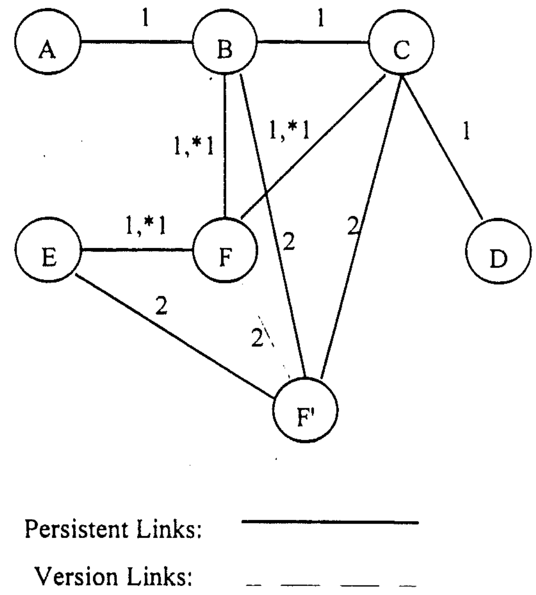


Figure 2: A hypertext graph with two versions

Assume that a change occurs in node C in version 1 of the model. Using the same procedure as before a new node C′ is created and the proper links connect the new node C′ with the nodes connected with node C, that is nodes B, F and D (Figure 3).

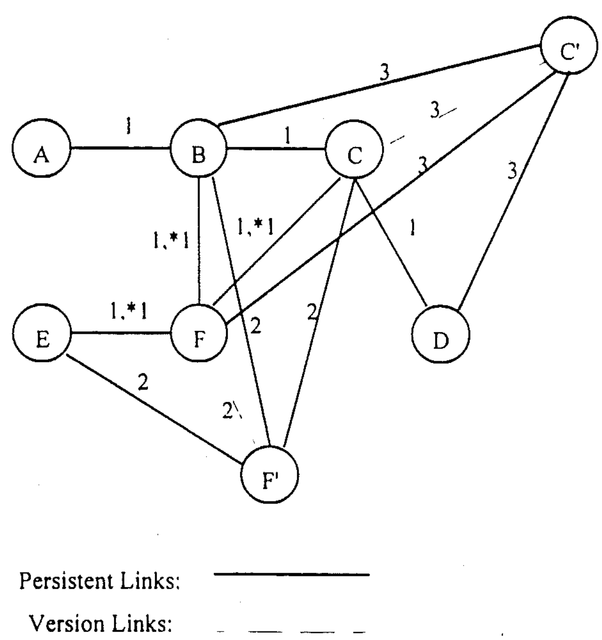


Figure 3: A graph with three versions

A problem that has to be solved in maintaining the history of the system is the non-linear ordering of the version numbers. For example, in Figure 3, version 3 is created from version 1 and although the order 2 less than 3 stands for the numbers 2 and 3, the same does not stand for version 2 and version 3 (actually they are independent from each other).

This partial ordering is defined by a rooted **version tree**, whose nodes are versions (1 through maximum number of versions) with version *i* the parent of version *j* if version *j* is obtained by updating version *i*. Version 1 is the root of the version tree. The sequences of updates giving rise to version *i* corresponds to the path in the version tree from the root to *i*. The version tree for Figure 3 is presented in Figure 4.

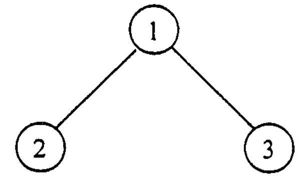


Figure 4: The version tree of the graph of Figure 3.

A more complicated example is presented in Figure 5. Figure 6 contains the version tree of the model in Figure 5.

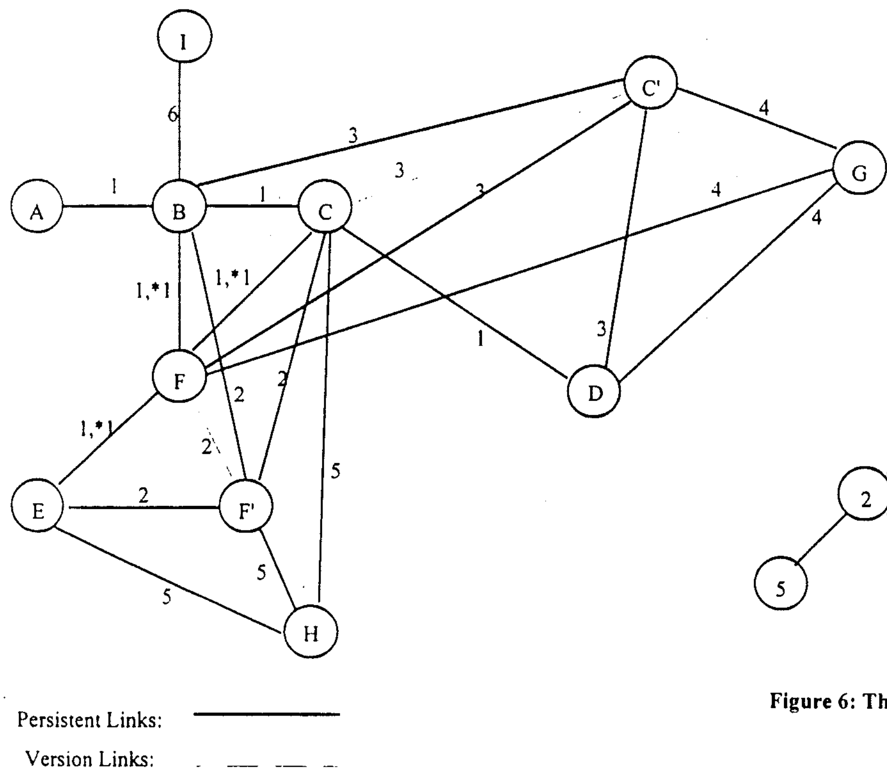


Figure 5: A more complicated example

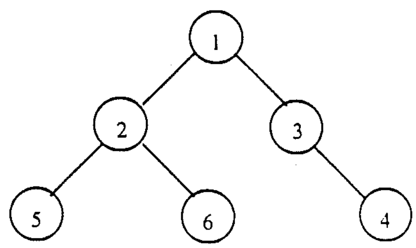


Figure 6: The version tree of the graph of Figure 5

In contrast to the model's complexity, the traversal of the network is quite simple. It actually follows the principles carried in current practise ([10], [20], [16]) enhanced with the special use of version tree.

From the moment a user has chosen the version he is interested in, the model can control the navigation between the nodes of the chosen version. By this we mean that a user is traversing only through nodes of the desired version, unless he wants something different.

In case a user is traversing a version, then he can only access nodes, connected with the current node, through links with version number equal or "less" than n. This ordering is defined by a preorder traversal of the version tree. These links should not be marked as updated. If there is an update mark on the link, then, if the number assigned to the mark is "less" than the current version number, the link is not available to the user. Otherwise (in case of equal or "greater") the node is part of the version and the link is usable by the system. Links with version number "greater" than n are not traversed, as they lead to a node belonging in a later version than the current one.

Consider again Figure 2. In version 2, the accessible links are the AB, BC, CD, BF', CF',

EF′. The BF, EF, CF links are not available, because they have an update mark with number 1 ("less" than 2). In version 1, links EF′, BF′, CF′ are not accessible, because their version number is 2 ("greater" than 1). All other links are part of version 1 (even the marked links are part of this version since the mark is equal to 1).

Here we should refer to the way we handle deletions. We do not allow the destruction of a node, because this would cause a version to be lost. When a user selects a node to be deleted, then the current version number is increased. The links that were used to connect the deleted node with the other nodes are marked as deleted. This means that the "deletion/update mark" field is set and is assigned the value of the latest version, the node appeared. The way we handle a deleted node, is similar to that of an updated node.

4.1 An Abstract Analysis of the Performance

Let us now compare the PH model with the data models used by other systems. We will also describe the possibilities that our model offers.

Till now existing systems used two different ways to manipulate versioning. The one was to have no versioning. The other way was to keep the different versions of a frame inside the same node ([17]). The user can access the most recent version of a frame and then he would search the frames of the node internally, in order to find any version he wants.

Studying this solution, we can say that it gives an answer to some problems, but this is not enough. There is a lack of an efficient way to answer queries of the form:

"Transform all the boldface characters to italics in the X version of the document stored in the system ([9],[12])"

where X is a version older than the current one.

In this case, the user should visit every node of the graph, find the frame that corresponds to the desired version and then make the changes he wants. This resides in the following problem: since every frame can be updated separately, there is the danger that the version numbers of the frames are inconsistent. So the user should use his instinct to find the right frames. This resides to a serious lack of system's reliability.

With the model we present here, we give a different solution to this problem. This solution does not require the direct co-operation of the user. Since a user has chosen the version he wants to access, then the persistent links lead him immediately to the corresponding version of each frame. This is done using the algorithm described above. That means that any system, using the PH model, has the ability to distinguish among different states (versions) of the system. *It actually provides a general view of every individual version of its states, supporting the handling of every version, as an entity.*

We should also notice that the ability to navigate through all or some versions of a specific frame is not restricted. The version links are used for this purpose.

Here, we would like to present the advantages and the disadvantages of the proposed model, concerning the usage of time and space.

The needs of space, compared to these of the solution used by other systems (multiple versions stored in the same node), are slightly worst. This is due to the existence of more links. But the space consumed by these links is not crucial. If the space used by the first solution is O(n), where n is the number of the nodes, in our solution this relation is not changed, because the number of the additional links is proportional to n. So the used space is also O(n). Another reason, that contributes to keep the space bound in O(n), is the fact that we allow a finite number of versions in the system. This number can be as large as we desire, but it should always be limited. A formal proof of these remarks follows from the proves of the time and space requirements of persistent data structures presented in [8].

The contribution of the theory of persistence is significant. If someone would like to save the history of a system, without using persistency's concepts, he would probably end up in the versioning control supported by other commercial systems. If he would like to enhance the system with the ability to update any version, he would actually end up in a system that keeps a complete copy of the structure for every instant (version) of the network. This resides in an exponential space cost. Persistence provides a space-reduction mechanism that allows every version to be saved on the original graph. This has a space cost approximately equal to the space required by only one version (which is always equal to O(n)).

5. Areas of Interest

Research areas are special areas, where a system supporting full versioning would be of great importance. Most scientists and researchers use common notebook features to collect the results of their work. The use of computer systems has given them efficient ways to organise the results of their work. Anyway the nature of scientific research requires back-tracking mechanisms, so that the researchers can easily review and correct previously done work. Old considerations should be re-evaluated and newly made observations should be added. The proposed model is intending to cover this need.

PH model, enhanced with persistent concepts, provides a very powerful backtracking mechanism. Any scientist-researcher can locate any previous version of his work. This version can be taken as a basis to develop a different course in his research. Updating features in the history of the data give researchers the ability to make a new version, with new considerations of their research's course. The ability to navigate through different versions of the same piece of data gives researchers the chance to estimate previous done work and locate possible mistakes.

Let's give two more examples for the use of a fully versioning support system. Assume that we want to have a hypertext system with historical data from the beginning of the twentieth century till now. It is obvious that a person interested for every event before the First World War would make a network with the available information. Another person interested in the music history of 20th century would do the same. Many of the data of both networks are the same, but only the organisation and the annotations would be different. A flexible hypertext system should be able to handle both cases, allowing the users to follow any of these networks.

With a similar way, assume that we have the documentation or the source code of a software product in a hypertext system. A part of this code is dependable of the environment the code runs (VAX/VMS, UNIX etc.). But a large part of the code may be the same for any environment. An efficient hypertext system should allow a user to follow both versions or remain in a desired one, or follow arbitrary one after the other. The PH data model guarantees the ability to do this with the minimum space and time cost.

6. Conclusion

Many systems are trying to enhance hypertext with versioning. Although the support of viewing any version of a system seems very common in commercial systems, the maintenance and update of system's history seems to be time and space consuming. The PH data model, using persistent issues, can reduce the time and space required for this purpose. This work is a first approach in using persistence for versioning in hypertext. A few modifications should be done in order to reduce the complexity of the model. This is definitely a part of our future work. But a major part of our work has to do with the implementation of this model and the actual testing in real life conditions.

References

[1] Akscyn R. M., McCracken D., Yoder E., *KMS: A Distributed Hypermedia System for Managing Knowledge in Organizations,* Communications of the ACM, Vol. 31, No. 7, pp. 820-835, 1988.

[2] Amann B., Cristophides V., Scholl M., *HyperPath/O2: Integrating Hypermedia Systems with Object Oriented Database Systems.* in Proc. of The Eight International Symposium on Computer and Information in Science, pp 709-720, 1993.

[3] Begeman M. L., Conklin J., *The Right Tool for the Job,* BYTE, Vol. 11, No. 11, pp. 255-268, October 1988.

[4] Campbell B., Goodman J. M., *HAM: A General Purpose Hypertext Abstract Machine.* Communications of the ACM, Vol. 31, No 7, pp 856-861, 1988.

[5] Croft W. B., Turtle H., *A Retrieval Model for Incorporating Hypertext Links,* in Proc. of Hypertext 89, pp 213-224, 1989.

[6] Crouch D., Crouch C., Glenn A., *The Use of Cluster Hierarchies in Hypertext Information Retrieval,* in Proc. of Hypertext 89, pp 225-237, 1989.

[7] Delisle N., Schwartz M., *Neptune: a Hypertext System for CAD Applications,* Annual Conference of the ACM SIG on Management of Data (SIGMOD), pp. 132-139, Washington D.C., 1986.

[8] Driscoll J., Sarnak N., Sleator D., Tarjan R. *Making Data Structures Persistent,* Journal of Computer and System Science 38, pp. 86-124, 1989.

[9] Fuller M., Mackie E., Sacks-Davis R., Wilkinson R., *Structured Answers for a Large Structured Document Collection,* in Proc. of the Sixteenth Annual International Conference on Research and Development in Information Retrieval, pp 204-213, 1993.

[10] Garg P. K., *Abstraction Mechanisms in Hypertext,* Communications of the ACM, Vol. 31, No. 7, pp. 862-870, 1988.

[11] Garrett N., Smith K., Meyrowitz N., *Intermedia: Issues, Strategies and Tactics in the Design of a Hypermedia Document System,* Proceedings of the Conference on CSCW, pp. 163-174, MCC Software Technology Program, Austin Texas, 1986.

[12] Glushko R., *Design Issues for Multi-Document Hypertexts,* in Proc. of Hypertext 89, pp 51-60, 1989.

[13] Halasz F. G., Moran T. N., Trigg T. H., *Notecards in a Nutshell,* Proceedings of the ACM, CHI+GI Conference, pp. 45-52, Toronto, 1987.

[14] Halasz F. G., *Reflections on Notecards: Seven Issues for the Next Generation of Hypermedia Systems,* Comm. of the ACM, Vol. 31, No. 7, pp. 836-852, 1988.

[15] Maurer H., Scherbakov N., Srinivasan P., *A New Hypermedia Data Model.* in Proc. of The Eighth International Symposium on Computer and Information in Science, pp 685-696, 1993.

[16] Nielsen J., *The Matters that Really Matter for Hypertext Usability,* in Proc. of Hypertext 89, pp 239-248, 1989.

[17] Prevelakis V., *Enhancing Hypertext Through Versioning,* Proceedings of the Greek Computer Society Conference, Vol. 1, pp. 285-297, Athens, May 1991.

[18] Salton G., Allan J., Buckley C., *Approaches to Passage Retrieval in Full Text Information Systems,* in Proc. of the Sixteenth Annual International Conference on Research and Development in Information Retrieval, pp 49-58, 1993.

[19] Tompa F. WM., *A Data Model for Flexible Hypertext Database Systems,* ACM Transactions on Information Systems, Vol. 7, No. 1, pp. 85-100, January 1989.

[20] Van Dyke Parunak H., *Hypermedia Topologies and User Navigation,* in Proc. of Hypertext 89, pp 43-50, 1989.

Developing and using documentation tools for Setext

David Martland
Department of Computer Science, Brunel University
Uxbridge, Middlesex UB8 3PH, United Kingdom
email: David.Martland@brunel.ac.uk

Abstract

The creation and maintenance of documents which are to be used in multiple modes, including printed texts, on-line documents, and hypertext, poses additional challenges for an author. This paper shows how, by the use of the markup language Setext, and appropriate tools, this effort can be reduced. The development and usage of documentation tools based on Setext, and in particular the creation of the Setext2latex conversion tool is discussed in some detail.

Keywords: hypertext, document maintenance, markup languages

Introduction

Interest in electronic document distribution systems and hypertext systems has increased considerably in recent years, and many documents are now distributed using a variety of methods such as Gopher, World Wide Web (WWW), ftp, or as files in a local file system. The format of the distributed documents may be various, including ASCII text, PostScript or .dvi files for printing and screen display, or as hypertext documents including embedded links. A common form of hypertext distribution is by files containing HTML (HyperText Markup Language) markup.

A motivation for the work described here is to simplify the writer's work when creating documents which are to be used in more than one of the text modes discussed above. This has been achieved by the adoption of a simple markup language, Setext, designed and developed by Feldman (1992), and by the use of tools for processing marked up text. Several tools have been developed, including the Setext2latex tool which converts Setext to LaTeX(Lamport 1986), and which is one of the major themes of this paper.

A further motivation is to remove the need for writers developing hypertext for WWW to use relatively complex markup languages, such as HTML, which obscure the text, and distract the writer.

Incidental side effects of the work discussed here are that a means of indexing documents easily, for both hypertext and printed documentation, has been developed.

The author's role

A writer's primary role should be to have ideas, structure them, and to express them in language. Producing text is necessary, though can be delegated to others. Many modern writers also produce their own texts, using computer technology. This may require them to type, edit, typeset, illustrate, print and publish their work. Publishing may simply mean making one or more files available to the intended readership over an electronic network.

Readers may like to have a document available in several forms - for example some readers prefer ASCII text files, often because these are easy to transmit in electronic mail, while others will have access to hypertext browsers, and may prefer to use searchable hypertext. Some readers may like to read well laid out text on the screen, while others may prefer to print the text, and read it conventionally. Figure 1 illustrates the modes which readers might expect to be made available for these different purposes.

Creating documents to satisfy each of these reader requirements increases the work of the author or publisher, and if document distribution is implemented by the author, this distracts from the primary roles discussed.

More detailed descriptions of authoring tasks are to be found in Brockmann (1990), and readers' views are in Nielsen (1990). Useful practical applications of presenting texts as hyperdocuments are given by Delany & Landow (1991), with technical aspects covered by Rada (1991).

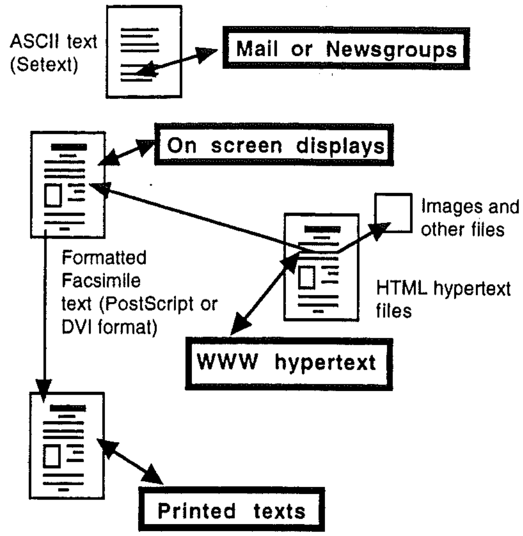


Figure 1: Document modes.

Solving the multi-mode problem

Ideally the solution to the problem of developing documents for distribution in multiple modes, is to use a single source form for documents, and to create all the other forms from this source.

This desire to handle several output forms from a single input document has led to the development of systems such as Texinfo, and the latex2html tool produced by Drakos(1993,1994a,1994b).

However, a difficulty with these solutions to the problem is that they require the author to learn a complex markup language. Where there is a need for intricate markup, as with mathematical articles, this will be tolerated by the author, but in many situations what is required is text which is well laid out, but which does not require the author to perform detailed type setting.

Setext

The Setext markup language allows a writer to generate text which is both readable in its ASCII source form, and which can be converted to other forms, such as PostScript for printing, or to HTML for distribution over WWW.

It is simpler to use than LaTeX, although without the fine control possible with that language, and the source text is much more readable.

Other means of text development include generating HTML using WYSIWYG editors, which are now becoming more common, and using word processors together with appropriate filters for converting the generated text into HTML format.

The use of Setext overcomes most of the problems discussed, and is truly multi-mode, although at the present it does not support mathematics typesetting well. Further, the additional time required to learn, and also to support the multiple modes of document in the generation process, is a relatively small overhead, when compared with other systems.

Basic Setext

The following table gives a brief introduction to Setext covering the essential features, mostly by example:

```
+++++++++++++++++++++++++++++++++++
  CORE SETEXT FEATURES
  +++++++++++++++++++++
  CONSTRAINTS
    <--------- 68 or fewer ---------->
     Lines of text should not
      exceed 68 characters.
  HIGHLIGHTS (only apply in flowing text)
      Bold          Underlined      Italic
     **Bold**     _Underlined_    ~Italic~
   **Bold words** _Underlined_words_ -
  QUOTED TEXT (written by another author)
     is introduced by a  >  in the first column:
         > This section of text is quoted
         > and this is a quote continuation.
  FLOWING TEXT
          has exactly 2 space characters
          at the start of each line.
          Lines of flowing text are joined,
          then word wrapped.
  PRE-FORMATTED TEXT
    Text which does conform to any other
    rule is pre-formatted, and should be
    laid out exactly as in the source file.
  COMMENTS
         .. this is a comment - comment lines
         .. begin with two periods.
         .. Some browsers may use comments
         .. as directives.

  BULLETS
         * This is item 1
         * This is item 2 -
           asterisk is in column 1
  HEADERS
         Main Header
         ===========
                      Section text ...
           Sub header
```

```
    --------------
        The underlining of headers
        should start in column 1
HYPERTEXT LINKS
        In flowing text, links_ for
        hypertext_anchors_ are
        indicated by a trailing underline,
        and matched up with
        special directives, as in:
      .. _links  FILE
      .. _anchor  http://anchor.url

 In the directive line, each link is
 matched with a leading underline,
 and the hypertext link is to the
 file or URL specified.
END OF SECTION
  The end of a section is denoted
  by $$ starting in column 1
END OF SETEXT
  A comment with no text terminates
  the Setext, as in:

    ..
    This is the end
+++++++++++++++++++++++++++++++++++++

ADDITIONAL FEATURES OF Setext2Latex

TYPEWRITER MODE (flowing text only)
   ++ typewriter text ++  Same font as
                          preformatted
                          text

DESCRIPTION LISTS
  + Description
     Body of description - start
       of description list indicated
       by + in column 1

INDEX SUPPORT (flowing text)
    =index=words=            words to be
                              indexed
    ==Capitalised==          Capitalised
                              index words

IMAGE SUPPORT (for hypertext)
  .. image  _imagelabel  URL of image file

FIGURE SUPPORT (for printed documents)

  .. figure _figurelabel  File  Caption
```

```
  Text Example in Setext
  =====================
  This is an example of text written
  using Setext_, a simple markup
  language. It includes both
  **hypertext links** and
  _other_emphasised_words_,
  which may be rendered in ~italics~.
  Besides this, the markup supports
  outlining, by means of

* Section headings
* bullet points

      Preformatted text is supported
              as in this section
                        here
> and quotations have their
> own style too, shown here
> with a leading > in column 1.

.. _Setext URL-to-Setext-description
```

Figure 2: A short Setext example

```
CITATIONS
  .. _citationlabel cite: Citereferences
+++++++++++++++++++++++++++++++++++++
```

Further information on Setext is given in Martland (1994).

Text examples

This section seeks to justify, by example, the benefits to the writer of using Setext for document creation.

Three versions of the same short piece of text will be given, which should demonstrate the relative ease of use of text in Setext form. The first of these (Figure 2) is also shown in its rendered form (Figure 3) - the results for the other two (Figures 4 and 5) are similar.

An argument can be made that the Setext example is clearer for the reader than the other two, and since writers also read their own texts, this should also help writers. The relative advantages of the three mark up forms will be shown in the next sections.

Text Example in Setext
This is an example of text written using **Setext**, a simple markup language. It includes both **hypertext links** and *other emphasised words*, which may be rendered in *italics*. Besides this, the markup supports outlining, by means of

- Section headings
- bullet points

```
Preformatted text is supported
              as in this section
                       here
```

```
    and quotations have their
    own style too, shown here
    with a leading > in column 1.
```

Figure 3: Printed version of Setext example

```
\section{Text Example in \LaTeX{}}
    This is an example of text written
    using
    \htmladdnormallink{\LaTeX{},
     URL-to-Latex-description} a very
    useful markup language.  It
    includes both {\bf hypertext links}
    and {\em other emphasised words},
    which may be rendered in
    {\em italics}.  Besides this, the
    markup supports
    \begin{itemize}
    \item       Section headings
    \item        bullet points
    \end{itemize}
    \begin{verbatim}
        Preformatted text is supported
                as in this section
                        here

   \end{verbatim}

   \begin{quote}
    and quotations have their own style too,
    shown here with their own environment
    delimiters
    \end{quote}
```

Figure 4: $\LaTeX$ example

```
<h1> Text Example in HTML </h1>

This is an example of text written
using
<a href = "URL-to-HTML-description">
HTML </a>, a markup language used
with the World Wide Web.  It
includes both <b> hypertext links</b>
and <u> other emphasised words</u>,
which may be rendered in
<i>italics</i>.  Besides this, the
markup supports
<ul>
<li> Section headings
<li> bullet points
</ul>
<pre>
    Preformatted text is supported
            as in this section
                     here
</pre>
<quote> and quotations have their
 own style too, shown here with its
 own HTML tags.
</quote>
```

Figure 5: HTML example

Relative Advantages

The relative advantages of the markup languages are shown below:

```
Setext:   Easy to learn
          Easy to read
          Easy to write
          Supports large documents
          Indirect hypertext support

Latex:    Good maths support
          Tables supported
          International character set
          Fine control of layout
          Indirect hypertext support
          Stable
          Widely used and available

HTML:     Direct support for WWW
          International character set
          Available, and very widely used
          Satisfactory graphics support
```

Relative Disadvantages

The relative disadvantages of the three markup languages are shown below:

```
Setext:   No mathematics support
          Not widely known, available
          No fine layout control
          Evolving
          Limited support for graphics

Latex:    Readability of marked up text
          Learning commands
          Awkward support for graphics

HTML:     Readability of marked up text
          Learning commands
          No fine layout control
          Large documents difficult to
              handle
```

We need to remember that in the future most computer users will not expect to have to be very knowledgeable about computer systems in order to use them, thus it is reasonable to expect that many users would prefer to use a simple markup language such as Setext, or to use a WYSIWYG tool, or to use a word processing package which they know, together with appropriate hypertext tools, in order to create printed documents and hypertext.

Balanced against this, writers will have a responsibility to create usable texts, and this will sometimes mean that writers will have to use more complex tools for certain applications. Nevertheless, many writers will find that producing documents using Setext should make the writing and maintenance task easier.

WYSIWYG tools

Several workers are developing tools which provide WYSIWYG editing. At the present time, many of these tools are not so easy to use as they could be, and if generation of WWW hypertext is the goal, many of them require the writer to be aware of HTML constructs. Further, many of these tools are too specific to hypertext generation for WWW, and do not address the problem of producing printed and hypertext documents from the same source. There is however, no obvious reason why a good WYSIWYG tool should not be developed, and this would certainly provide strong competition for Setext. Such a tool would need to have a standard representation for text, and would also need to maintain information about the printed appearance of the text. WYSIWYG tools take a different route to the production of documents, which may ultimately turn out to be as effective as the use of Setext proposed here - however there appears to be no good reason why such tools could not coexist with Setext, and indeed Setext production itself could benefit from the use of WYSIWYG editing tools. At present this is only hinted at with the browser tool Easy View developed by Eyler (1993), which is read only.

At present, many hypertext authors are writing directly in HTML - this is the hypertext equivalent of programmers writing in assembler code - HTML is usable, but it is an awkwardly inappropriate language for writers to work in, though this is often

denied by computer literates who don't appear to understand the problems faced by others less fascinated by the workings of computer systems than themselves. Sometimes HTML allows greater control over the text, and where visual impact is required, the use of image files embedded into the text may be achieved effectively by the use of HTML. Such customized texts are appropriate for front pages, where striking appearance may be important, but for many documents which may be accessed from such front pages, more conventional text, with relatively few illustrations, may be more usual, and this will be easier to create and maintain by the use of tools such as Setext2latex or latex2html, and there will almost certainly be a productivity gain in the use of such tools for text production.

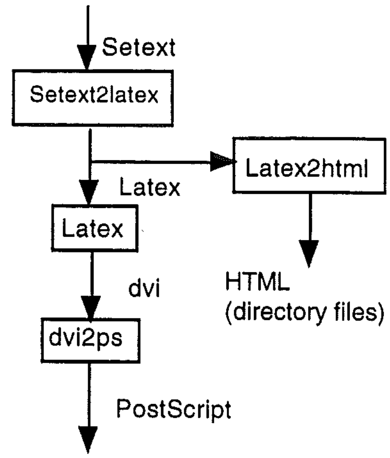


Figure 6: Setext data flow.

Setext2latex

Setext2latex, or s2l is a tool which converts from Setext to LaTeX. This allows a writer to generate text simply, and then prepare it for on-line display or printing.

Writers who wish to use the full power of LaTeX may not wish to use Setext, although it is useful for obtaining draft versions of LaTeX marked up files, but many writers should find that working wholly in Setext is much simpler because of the use of the visually meaningful or unobtrusive mark up tags. A further advantage of using Setext is that Setext2latex almost always generates valid LaTeX, thus avoiding the difficulty of having to "debug" the LaTeX source.

By using Drakos' latex2html tool, the generated LaTeX can be converted to HTML, and this allows writers to generate hypertext.

Figure 6 shows the file types supported by the combination of Setext2latex and latex2html.

An alternative tool developed by Sanders (1993) allows direct conversion of Setext to HTML. This has the advantage of simplicity, but does not provide such a comprehensive level of support for the writer.

Setext2latex provides most of the facilities of the "core" Setext language, and in addition also provides support for:

- specialized document types
- inclusion of pictures and images in hypertext
- inclusion of figures in printed text
- index generation (automatic in hypertext)
- limited access to LaTeX commands
- extended footnotes
- automatic conversion of network addresses

Specialized documents include journal articles or reports, and letters, and can easily be extended to memos, faxes and other commonly used document types. For printed documents it is also possible to maintain limited control over font size and line spacing, and two column mode is also supported.

If documents can be written wholly in Setext, then maintenance is eased considerably. This is not always possible, and it may be necessary to make direct use of LaTeX features. If this can be done without having to make the LaTeX source the primary source, this will be beneficial, and this can often be achieved by the use of included LaTeX files using the \include or \input commands. The two features of LaTeX which academic authors will value are:

- mathematical typesetting
- use of bibliographic database

The current Setext2latex utility allows partial access to LaTeX features by optionally changing the meaning of the characters 4

```
\  {  }
```

to allow embedded LaTeX, and thus the inclusion both of some mathematics, and citations using the \cite command. Thus it is possible to produce academic articles with citations and reference lists with no more difficulty than using LaTeXand BIBTeX directly.

Access to LaTeX commands is only partial, since some commands conflict with Setext constructions,

but otherwise there is no restriction. Access to mathematics via this mechanism is feasible, though tedious, since the $ and $$ notations for maths and displaymath modes are disabled. Particular problems arise with the use of subscripts, since these would appear in LaTeX as, for example:

```
x_subscript
```

but this is treated as a hypertext anchor "x" followed by text "subscript".

Full mathematics capability will be added in a future release.

A recent modification to the s2l software allows citations and references to be included without the need to enable LaTeX commands. This simplifies the text creation, although the author still needs to create the bibliographic data base for BibTex.

Indexing is simple, and requires words to be included in the index to be surrounded by equals signs, as in this example:

```
This =word= should appear in
the index, and so
should =this phrase=.
```

All index words are converted to lower case, unless specifically indicated by the use of an == combination, as in:

```
The ==Roman=Empire== gradually gave
way to the ==Byzantine=Empire==.
```

Setext2latex also expands certain character strings which represent meaningful network entities, such as mail addresses, which are conventionally of the form:

```
<mymailid@mycomputer>
```

and network addresses which are identifiable by a form such as:

```
http://address      ftp://address
gopher:// address   telnet://address
tn3270://address
```

These can be used in text to give quick access to network resources, and additionally, already existing text, such as email messages can be converted into active text using this feature. This provides a quick way of generating address lists and hotlist of network resources.

Developing Setext2latex

Setext2latex was developed using the Perl programming language due to Wall & Schwarz (1991), and this has been found to be a very quick. though somewhat unusual method of developing the code. This operates mostly by the use of regular expression string matching on a line by line basis. Even allowing for the fact that the generated code is not optimized for speed, and must scan each line several times, the performance of the converter is good, typically converting one page of text per second on a networked Sun workstation.

Learning sufficient Perl to generate the converter has been relatively slow, and the "write only" nature of many available programming examples has presented difficulties. The converter has been written with use of subroutines even for very small fragments of Perl code - this has had benefits in aiding understanding and also for code re-use. Readers of the code will still have to contend with dense fragments such as:

```
s/(^[\s]+)/  \1/; # insert 2 spaces
                  # if first
                  # character non-blank
/^  [\s]+/ && do {&doflowing;};
        # if 2 spaces followed by
        # non-blank do flowing mode
```

However, the use of Perl has undoubtedly enabled very rapid development and testing of new features, and since the performance is adequate there seems to be no immediate need to recode it. Particular problems have arisen with the use of special characters used by LaTeX, and these have been solved by translating them into long and unusual character strings, while other processing is carried out, and then finally converting back into appropriate forms. A more conventional parsing scheme would allocate tokens to these characters, and would probably be based on a more efficient deterministic recursive descent parser.

Use of Setext

Setext use is spreading slowly, and it has not been possible to conduct a comprehensive survey of users. Versions of the Setext2latex tool have been used to generate a significant number of texts for printing, including course documentation, student example sheets, and longer articles, and indeed, for preparing this paper. Some Setext documents are now available on the Internet. The TidBITs on-line magazine

is available in Setext form, and issues of this have been used for testing the printed output. Further tests have been carried out by using the html2setext tool developed by Pam (1994) to generate Setext from networked hypertext, and this has been useful in detecting and correcting program bugs.

Using Setext2latex

Generally Setext2latex is easy to use, though some documents, such as journal articles, are more difficult to process. Experience gained while writing this paper and other articles will be discussed briefly here.

Most of the preparation of this paper was carried out in Setext, although a disproportionate amount of time was spent on incorporating the diagrams (using the epsf package), and constructing the bibliography. These appear to be common difficulties with papers written using LaTeX, and clearly need of further attention. Several of the diagrams were produced using a commercial package to generate encapsulated PostScript. However, care needs to be taken to incorporate diagrams correctly, since the processing of .dvi files generated by LaTeX for the inclusion of the PostScript, appears to differ between different systems, and this can cause major problems.

The text examples were set as LaTeX figures, using the \minipage environments, and although Setext does not support these, they were easily achieved by inserting the commands into preformatted text, and then post-processing the generated output file. The only other LaTeX post-processing performed on this document relates to the use of the Harvard bibliographic style, which is not fully compatible with the standard BibTex commands, and was relatively simple to carry out. The post processing was very limited, and thus does not violate the claims made for easing maintenance. One other area where the use of Setext could be expected to offer improvements over the use of LaTeX is in spelling checking. Unfortunately, this depends on the spelling checker used – the Unix spelling checker works well, but some other checkers work poorly because of the additional Setext markup.

A final comment - it is noticeable that Setext files, since they are treated as plain ASCII, are significantly smaller than the files maintained by typical word processing packages, and this may benefit users who wish to maximise their use of storage space.

Generating and using hypertext

Hypertext is gradually finding its way into the curriculum at many universities. Setext2latex has greatly enhanced the ability to produce on-line hypertext within a limited time scale. Course notes are now available for many courses using hypertext based on HTML. Several courses at Brunel now have material distributed via WWW, and this has led to some reduction in the need for printing. The use of Setext for the development of documents for on-line and printed forms has reduced development times significantly – such documents were previously produced using LaTeX.

Although the facilities for creating on-line hypertext using HTML are useful within the context of courses, a major use for documents on-line is as on line "facsimiles", which are PostScript versions of printed documents, which students can view and print at will. Essentially Setext documents can exist in four different forms:

- Source text, containing visible mark up, which is readable, and can also be used in a text search.
- Rendered text, output on a printer, and distributed on paper.
- Rendered text, held as PostScript "facsimile" files, and distributed either as plain files, or by use of the Gopher or WWW interfaces.
- Hypertext, held as sets of HTML files, which allows users to browse, and also perform text searches.

It is very easy to make and distribute such documents, so that short printed documents can be made in a short time, ready for printing.

On one occasion when on-line facsimile versions of documents were made available, students had already printed their own copy from the on-line sources before receiving a printed hand-out, yet the document had only been produced two hours earlier.

The use of PostScript (or .dvi) facsimiles in electronically distributed documents appears to be very useful for readers, and the use of such a format should not be ignored by hypertext authors. Where HTML based WWW hypertext is to be generated, it would often be helpful if links to such facsimiles were provided from the hypertext versions.

Conclusions

This paper has presented a brief description of Setext and the Setext2latex converter, and a few of the factors in its development. Further, it has been shown that a simple mark up language, Setext, is capable

of being used as a single source language for both printed documents and hypertext. Emphasis has also been placed on the fact that use of such a simple system will permit writers to concentrate on language and concepts, and this should prove helpful to many who find the use of computer technology difficult or confusing. With its limited number of constructs, Setext imposes a very small cognitive burden on the writer. While better results may be obtained in some situations by the use of other tools, such as the use of LaTeX for mathematical typesetting, or using raw HTML for tighter control of WWW documents, in many practical situations, the ease of creation and use of Setext source documents will prove to be very cost effective. Alternative tools, such as WYSIWYG editors, are considered to be immature at this time, though have a promising future.

Both Setext, and the Setext2latex converter program are largely system independent; the former since it does not depend on the features of any single system, and the latter since both its implementation and target languages are widely available on many currently available hardware and software systems. A port of the converter to Apple Macintosh computers took half a day, including installing the Perl system.

Conversion to HTML is also largely system independent, because the latex2html converter is also written in Perl.

Acknowledgements

Lastly, thanks are due to Ian Feldman for many helpful suggestions, and to Tony Sanders for doing the original Setext to HTML converter without which Setext2latex would not have been produced. Nikos Drakos provided assistance with his latex2html program, and Andrew Pam's html2setext program and data files were also very useful in the testing phases. Thanks are also due to Brunel University for the use of its computer systems, and to Professor Ray J. Paul and my other colleagues for help during the development period.

References

Brockmann, R. (1990), *Writing better computer user documentation, from paper to hypertext*, Wiley, New York.

Delany, P. & Landow, G., eds (1991), *Hypermedia and literary studies*, MIT Press, Cambridge, MA.

Drakos, N. (1993), 'Text to hypertext conversion with latex2html', *Baskerville* **3**(2), 12–15.

Drakos, N. (1994*a*), 'From text to hypertext:a post hoc rationalisation of latex2html'. Available as http://cbl.leeds.ac.uk/nikos/doc/www94/www94.html.

Drakos, N. (1994*b*), 'The latex to html translator'. Also available as an electronic document as: http://cbl.leeds.ac.uk/nikos/tex2html/doc/latex2html/node7.html#pub.

Eyler, M. (1993), Closing the gap between outlining and hypertext, *in* 'Proceedings of the Workshop on Intelligent Hypertext, CIRM93', Arlington, VA. Article also available from <eyler@firat.bcc.bilkent.edu.tr>.

Feldman, I. (1992), 'Valid typotags table'. Available as: http://www.bsdi.com/setext/ typotags.txt.

Lamport, L. (1986), *Latex*, Addison-Wesley, Reading, MA.

Martland, D. (1994), 'Basic information about setext'.
Available as http://http2.brunel.ac.uk:8080/~csstddm/setextinfo.html.

Nielsen, J. (1990), *Hypertext and hypermedia*, Academic Press, Boston, MA.

Pam, A. (1994), *HTML to structured enhanced text filter*, Electronic document, available as http://www.aus.xanadu.com:70/0/sc/html2setext.c.

Rada, R. (1991), *Hypertext: from text to expertext*, McGraw Hill, London.

Sanders, T. (1993), 'Setext information'. Available as http://www.bsdi.com/setext.

Wall, L. & Schwarz, R. (1991), *Programming Perl*, O'Reilly & Associates, Sebastopol, CA.

InterJournal: A distributed refereed electronic journal

J. Redi and Y. Bar-Yam
ECS Dept., Boston University, 44 Cummington St., Boston, MA, 02215

Abstract

InterJournal is a refereed Internet-based journal that is accessible through the World-Wide Web. The articles themselves are distributed throughout the Internet. They may be stored by the authors "at point of origin" or by arrangement with colleagues or in Internet-accessible databases. The central journal database consists of abstracts, comments and relevant manuscript information including the Internet address of the original article. HTML forms are used to execute author/referee registration, manuscript submission and revision, referee reports, comments, and correspondence between authors and referees. The hierarchy of subjects within the journal allows a manuscript to be simultaneously submitted to, refereed and then accessed by several readerships, effectively making *InterJournal* a collection of interlocking journals. *InterJournal* can be accessed through the URL "http://dynamics.bu.edu/InterJournal".

1 Introduction

The World-Wide Web (WWW) initiative is a project that began at CERN as a method for physicists to share formatted data and results across the Internet (ftp://info.w3.org-www-doc/www-for-hep.ps.Z). The basis of this system is a standard hyper-text markup language HTML which enables access to files stored at remote locations and identified by Universal Resource Locators (URLs). The rapid growth of the World-Wide Web has expanded the possibilities for professional communication beyond that of print journals, including the advent of refereed electronic journals. Typical electronic journals receive Postscript or TeX formatted manuscripts through electronic mail or FTP. The standard reviewing process is then conducted through electronic mail. Finally, the accepted papers are included in the World-Wide Web (WWW) pages of the electronic journal. Some of the available electronic journals are listed at the URL "http://info.cern.ch/hypertext/-DataSources/bySubject/Electronic_Journals-.html"

In this paper, we describe the recently launched *InterJournal* that differs from standard electronic journals by utilizing the shared resource structure of the Internet. By using URLs to identify manuscripts across the World-Wide Web, it is possible to automate and drastically reduce the time necessary for the standard cycle of submission, comment, revision, refereeing, and ultimately acceptance or rejection. At submission, the authors make their manuscript available on the WWW. Abstracts, comments, and referee reports are maintained centrally, but even these are cataloged automatically by URL. Referee reports and comments are almost instantly accessible to the author. The distributed nature of the journal allows it to be virtually unlimited in size and scope. The electronic nature of the journal allows it to be quickly searched by a reader for only the relevant manuscripts.

2 Discussion and Motivation

InterJournal relies upon authors to store and make their manuscripts available on the Internet. This enables a number of changes from conventional publication.

2.1 The "manuscript"

One of the major recent advances in print publication is the introduction of color. *InterJournal* takes this further by enabling all possible data formats that can be stored on a computer and transferred through the Internet: text, color figures and pictures, computer programs, raw data, video, audio and documents that are structured by hyperlinks. In addition to the enhanced communication, *InterJournal* enables new opportunities for what, in effect, become distributed collaborations. Such collaborations are a major novel feature of *InterJournal* publication. Since the format and amount of information is virtually unlimited, authors are free to include materials that were not previously feasible. For example, it is possible to publish large amounts of data for others to interpret and use in their own research. Programs for simulation and interpretation can also be attached. Communication and continuing research between parties that wouldn't otherwise have such access to each other's research methods and tools are encouraged.

The potential for collaboration is measured by the proliferation of postings on the Internet. The binding of data and programs to publication and a systematic subject hierarchy has advantages in effective communication. Comments and refereeing can provide essential feedback and information to other interested researchers. Comments can also contain the results of subsequent use.

As authors become more familiar and comfortable with this new medium, we expect a shift from essentially an electronic form of a traditional paper journal, to a new level of scientific collaboration, publication and review.

2.2 Refereeing

In recent years electronic databases of preprints have become standard in a number of fields. These databases provide access to manuscripts before they are in print. *InterJournal* enables public access to manuscripts immediately upon submission. However, at the discretion of the author, access may be limited and a more traditional refereeing can be done initially using editor selected referees. Refereeing serves three distinct purposes:

1. To evaluate manuscripts so as to determine category of acceptance or rejection.

   The manuscript categories are related to the importance of the article and the related professional recognition. The categories parallel existing journals: Letter, Review Article, Article, Brief Article and Report. Acceptance of a manuscript enables the author to refer to the manuscript as published (accepted) in an archival journal. The use of categories of publication in *InterJournal* avoids publication delays due to submission of a manuscript, followed by rejection, and the need to resubmit to a different journal. The manuscript should rise or fall to its level of recognition within *InterJournal*. Refereeing still provides a means of filtering articles and labeling them by quality for the benefit of both author and reader. A "News item" category enables the announcement of conferences, workshops, books, etc. of relevance to the readership.

2. To direct manuscripts to the interested audience.

   It is generally recognized that the number of articles published, even in a particular field, greatly exceeds the capacity of anyone to read them. This is often attributed to an abundance of poor quality work. However, there are published a great number of articles that are of high quality. Everyone must limit the number of articles he/she can read. An essential function of refereeing in *InterJournal* is to properly identify the subject area of the manuscript - a first and significant step towards enabling the appropriate audience to be reached by a manuscript.

3. To provide corrections and advice to the authors of a manuscript.

   Confidential remarks attached to a referees report are communicated directly to the author of the manuscript. The author may make use of these to revise the

manuscript or may reply to the referee privately. This function of the refereeing process - to provide feedback on a manuscript - is maintained.

There are also several changes in the mechanism by which refereeing is done. Unlike other journals, in *InterJournal* refereeing can be done by any qualified referee who accesses the manuscript. This diminishes the common difficulties that arise in identifying appropriate referees. Using the received reports, the acceptance, category of publication, and subject areas of publication are determined by the editors.

InterJournal also maintains a more traditional method of refereeing as an option to the authors: An author can request that only editor selected referees have access to the manuscript during an initial refereeing period. This is a more conventional refereeing process and may be selected in order to receive referee comments before general exposure. By convention this also maintains confidentiality of the manuscript. If this option is selected, the author is also required to suggest at least five referees. An author may also indicate referees not to be selected for reasons of conflict of interest. However, the manuscript is not accepted until the authors perform a manuscript revision that allows open refereeing. Acceptance is determined after the open refereeing period.

Finally, both abstracts and hyperlinks to rejected manuscripts are maintained in the database - with the status "Rejected". This enables an author to establish a record of unpublished work. There exist varying opinions about the potential value of manuscripts rejected by referees at a particular time. Readers of *InterJournal* can choose whether to include such articles in their searches. Thus acceptance certifies professional merit by peer-review rather than enabling or restricting access. This only applies to articles rejected on professional grounds. The editors will summarily remove any "Junk" or inappropriate submissions.

2.3 Comments

Many journals provide a mechanism for brief comments that are directly related to a particular manuscript. This enables improvements, criticisms or additional references to be directly associated with a particular article. *InterJournal* treats comments as an integral part of the journal. The same page that gives access from the journal to the original manuscript also has hyperlinks to the comments on it. Comments are treated similarly to manuscripts and can be commented upon, refereed, and classified in the database.

2.4 Multiple virtual journals

One of the central problems with the large number of paper journals that are published is that each journal addresses only a small segment of the research community. Most research is of interest to more than one such segment. This is particularly true in the rapidly expanding areas of interdisciplinary research. *InterJournal* acts not as a single journal but as a number of interlocking journals. Articles that are submitted can be simultaneously refereed and accepted (or rejected) in more than one of these virtual journals. The different journals are identified by subject area. However, all of the subject areas are part of the same global hierarchy and the manuscript information of all of the different subject areas are stored in a common database. To access a particular area (i.e. journal) a reader specifies the topic of interest in a search.

The standard model of paper scientific journals is limited by the economic bounds of number of pages, number of manuscripts, and the effort necessary by administration to oversee all the editorial communication. *InterJournal* relies on the distributed and electronic nature of the WWW to automate many of the responsibilities and overcomes the inherent bounds of material space.

The centralization of the journal database also serves instead of electronic indexes of journal articles. Central electronic indexes serve readers. The advantages of the InterJournal index apply both to authors (for submission) and readers (for access). As stated previously *InterJournal* allows for a manuscript to be

submitted in multiple areas for simultaneous review.

Submission of manuscripts to different journals is accomplished by specifying subject areas at time of submission. The referees may change the "journals" in which the manuscript is eventually published by their referee reports that recommend subject areas for acceptance. The subject hierarchy (thesaurus) enables two different kinds of distinctions: topic and level of general interest. This is similar to paper journals that can be highly specialized or address a more general audience.

2.5 Archival database stability

InterJournal is designed to be a persistent archival resource. The ongoing reduction of library storage and the proliferation of journal titles, has led to a crisis in their archival value. Electronic access is already more universal that of many professional paper journals. The consequent archival value of electronic publication may rapidly exceed that of conventional journals.

InterJournal maintains the content of manuscripts through a scheme of manuscript retrieval and generation of a unique checksum. When a manuscript is submitted or revised, *InterJournal* retrieves the file and performs a checksum on the manuscript to discourage authors from changing the manuscript without making an official revision. This also allows *InterJournal* to maintain backups on secondary storage of these manuscripts, so that in the event of an author's manuscript going off-line, the contents of *InterJournal* will not be compromised. When the number of articles justifies, additional backups, multiple access sites, and archival mechanisms will be established.

The manuscript is defined as the document located at the URL specified in the submission form (see below) and one level of HTML reference from it. Authors are instructed to include all indirect references in the primary document which may be a cover page. URLs in the second level are allowed as references or additional information pointers, but are not considered part of the manuscript. *InterJournal* takes no responsibility for maintaining the validity of URLs beyond those of the manuscript.

InterJournal provides the important service of maintaining indirection for reference to scientific papers on the Internet. When authors move or need to change the location of their manuscript, it is only necessary to inform *InterJournal* of the changes. If all references to the article are made through the abstract page, it will not be noticed if the manuscript moved in actual location. This solution to the Internet volatility problem can be likened to sending postal mail by writing the recipient's name on the envelope. The post office would be responsible for maintaining a list of addresses, and a change in address would only require notification of the central database.

2.6 InterJournal Editors

The *InterJournal* editorial staff consists of area editors that are responsible for a particular section of the subject hierarchy. At present article submission is accepted in the following areas:

- Complex Systems (Editors: T. Toffoli, MIT, B. Boghosian, Boston University)
- Polymers and Complex Fluids (Editor: Y. Rabin, Bar-Ilan University)
- Genetics (Editors: C. L. Smith, Boston University, P. Pevzner, Pennsylvania State University)

In addition, there is an editorial advisory board consisting of:

- B. Alder, Lawrence Livermore National Laboratory
- C. H. Bennett, IBM T.J. Watson Research Center
- C. R. Cantor, Center for Advanced Biotechnology, Boston University
- E. Hartmann, Tufts University and Newton Wellesley Hospital
- J. Kagan, Psychology Department, Harvard University
- M. Kardar, Department of Physics, MIT

- C. Langton, Santa Fe Institute
- M. L. Minsky, Media Laboratory, MIT

The authors of this article are responsible for system administration and development (J. Redi) and editorial management (Y. Bar-Yam).

3 Summary of InterJournal Forms

HTML forms enable the communication between the remote clients (the readers and authors) and the server (the InterJournal database manager). The following sections describe each of the relevant forms.

3.1 Registration

All authors submitting manuscripts, comments and referee reports to *InterJournal* are required to register by use of the registration form. It is not necessary to register to search, access and read manuscripts. The registration form prompts the author to choose an Author ID and a password. The Author ID is a short, unique way of identifying users, analogous to the login ID chosen on a shared computer. The password enables confidential correspondence between authors and referees and maintains security. In addition to the ID and password, the registration form also contains fields for name, address and email, for external correspondence.

A unique feature of registration with *InterJournal* is the ability for an author to specify a Home-Page. This is a URL that is accessible through any article submitted by the author or from the list of *InterJournal* contributors. It is typically an HTML form constructed by the author that contains information and references to the author's current research.

3.2 Searches

One of the most important features of a journal of such potential size and diversity is the ability to do searches. Searches may be performed through one of two forms: search or browse. Both allow a reader to specify the type, area, or submission date of manuscripts they wish to see. The browse form is simpler and has more limited options. It should encourage beginning users and promote "browsing" the current on-line manuscripts. Additional search engines, such as author customized expert systems will be introduced when justified by the number of articles in the database. When a successful search is performed, the user will be presented with a list of titles, authors and submission dates that match the criteria of the search. This list is in hypertext so that clicking on a reference fetches the abstract page for that manuscript (see Section 3.4).

3.3 Manuscript submission

The manuscript submission form is filled in by an author to submit or revise a manuscript. This form transfers all relevant information about the manuscript to the InterJournal database manager. The mandatory fields include author IDs, contact author, title, abstract, the location (URL) of the manuscript on the WWW, and the subject areas that the author feels the manuscript belongs in. It is possible also to specify the manuscript category (Letter, Review Article, Article, Brief Article or Report). The subject areas and category that are specified by the author are only a recommendation. The final subject areas and category are determined by the referees. Manuscripts may be submitted with two options relevant to refereeing - anonymous, so that authors are not identified during refereeing, and editor selected referees, for initial limited access refereeing period. A list of suggested referees can be submitted. For editor selected refereeing such a list is mandatory.

3.4 Abstract page

An abstract page found by a search (see Section 3.2) contains links to the original manuscript, the authors' information, a comment/referee report form, and public comments on the manuscript. If the manuscript is a revision, there will also be a link to the abstract page for the previous version.

3.5 Comment and referee report submission

When a user clicks on the "Comment/Referee Report" button of an abstract page, a preformatted comment or referee report form appears. The form contains the manuscript number, the first author and the title of the manuscript. The use of this form as a comment or referee report is achieved by selecting options that specify whether the contents are "public" or "for authors only" and "anonymous" or "author identified". Referee reports are stored in files that can only accessed by the authors of the manuscript through the correspondence form (Section 3.6). Comments are treated similarly to manuscripts. An abstract page is created. The major difference between a manuscript's abstract page and a comment's abstract page is that the text of the comment will be listed instead of an abstract. An author has the option to store an extended comment on their own machine and submit an abstract of that comment as well as a URL to the comment's text. For either short or extended comments, a reference is placed on the original manuscript's abstract page, enabling readers to have direct access.

3.6 Correspondence

The correspondence form is accessible by an author or referee using the ID and password that have been registered with *InterJournal*. Using this form an author or referee can perform most of the functions that are user specific. These functions are:

1. to change user information such as address, email, etc.
2. to generate a search for submitted manuscripts that match the subject areas that the user specified during registration for refereeing.
3. to view the comments and referee reports made on his/her submitted manuscripts. The author is presented with a list of the manuscripts that he/she has submitted. Each entry is a hyperlink to a list of all the comments and referee reports on that manuscript. The comment and referee report list is a list of hyperlinks to the actual comment and referee report pages. Through these comment and referee report pages, the author can submit replies.
4. to view replies that have been made to his/her comments and referee reports. The author is presented with a list of all his/her comments and referee reports. Each of the entries is a link to a list of all the comments and referee reports made on that particular submitted comment, or a list of replies made to a referee report.

4 Issues Raised by InterJournal

- *What happens if nobody accesses the manuscript for refereeing?*

  As with paper journals, the editors are ultimately responsible to arrange for refereeing. To encourage refereeing we are considering a system that would require that manuscript submission must be balanced by a number of (say three) referee reports.

- *How is uniformity of appearance of manuscripts guaranteed?*

  There are general style sheets that recommend the format for *InterJournal* publication. Manuscripts are treated by the journal similar to a "camera ready" format print publications.

  The appearance of manuscripts and their readability affects the benefit to the audience. Thus, referees should take appearance into account in their reports. Aside from the recommendations and the actions of the referees there is presently no central control over manuscript appearance. Since authors are responsible for preparing manuscripts the appearance is a reflection on the care exercised by the authors.

  Electronic publication allows for the retention of older manuscripts in the same format, subject index, and location as the

most recent submissions. As manuscript file formats change, converters become available and migration to a new format is made as painless as possible. Under certain circumstances authors may be required to perform such conversions.

5 Discussion of Implementation

5.1 External representation

The InterJournal database manager addresses two main types of entities: authors and manuscripts. Within the manuscript definition is included revised manuscripts, comments, revised comments, and referee reports. When an author registers with *InterJournal*, two files are created. One file is an information file that contains the author's real name, email address, encrypted password, and other personal information. The other file is an activities file that contains, in HTML, a list of all the manuscripts that the author has submitted. An author can check on the status of his or her submitted manuscripts and retrieve the activities file (see below) of any of their manuscripts.

The password that is stored in the author's information file is encrypted with the standard DES encryption scheme present on most Unix systems. It is relatively secure. Even if someone gains access to these files the password would not be readily known. This security is critical to proper operation of *InterJournal* since it enables the journal to restrict access to private correspondence between authors and referees. It is also the only way of verifying the authors identity for comments and referee reports, and therefore has bearing upon the acceptance scheme for manuscripts.

When a manuscript is submitted, two files are created, an abstract file and an activities file. The activities file serves a similar purpose to the author's activities file. It contains a listing of all the revisions to that manuscript as well as all the comments and referee reports made to each of these revisions. The file contains HTML links, so that when viewed, any of the revisions, comments, or referee reports can be viewed by clicking on the appropriate reference. The revision of manuscripts without notification of the journal is prevented by checksums calculated at the time of submission.

InterJournal also keeps track of manuscript submissions with a central database file. This file contains an entry for each manuscript, and comment. This database can then be searched by submission date, category or area of interest. This file is a flat text file that allows for regular expression searches. When its size warrants, a hash table on various keys will be implemented to increase search speed.

Since there are potentially many possible simultaneous read and write combinations, a method must be used to ensure consistency within the writable files, most importantly the central database file. For this purpose, the exclusive read and exclusive write access privileges from Unix's open commend were used. Some variations of Unix, including IBM's AIX, contain additional provisions for locking writes but allowing reads to a file from multiple programs. Using this the InterJournal database manager attempts to open files exclusively when writing. If a read or write attempt fails due to another process' file activities, the process will keep trying to gain control of the file until a certain time-out expires. Since most of the files are short, the amount of time any process will have to wait to gain control of a file is expected to be short in comparison to latency in sending documents across the Internet.

5.2 Internal representation

In building *InterJournal* it was necessary to build a set of server programs to interact with client queries from the remote users. These programs are responsible for maintaining the databases of manuscript information, author information, and their activities. When a client submits data contained in an HTML form, this data is passed to a server program through it's environment variables. The program acts on the data, modifies the database, and communicates to the client through it's standard output channel.

The internal implementation of *InterJournal* is based upon a hierarchy of document

types implemented in C++, as shown in Figure 1. Each of these document types acts differently within the InterJournal database, yet they still contain similar critical information. The most basic type is called the Document. This class does not exist, but is a virtual class that is a common basis for all the other document types. It contains the list of authors who have written the paper, the contact author for correspondence, the location of the manuscript, the date the manuscript was submitted and a pointer to the abstract file. Each of the other document types contains additional information and functionality.

The substructure and hierarchy shown in Figure 1 is readily implemented using an object-oriented language such as C++. Another benefit from object-oriented programming, besides ease in creating hierarchies, is the use of virtual functions to enable runtime decisions while hiding the details, via abstractions, from the programmer. An example would be the type of abstract page that is generated. A comment or a revised manuscript's abstract page should have a link back to the original document, but the original manuscript should not. The program doesn't know which type of abstract should be written but sends a message back to the object that informs the data type that it's own type of abstract is to be written. To program the dependences of each of the types of documents that is used would add a great deal to the complexity of such simple functions as revising a comment. Using virtual functions, we are able to hide these types of decisions from the client programs, which allows additional document types to be seamlessly integrated into *InterJournal*, as well as making additional client programs simple to write and debug.

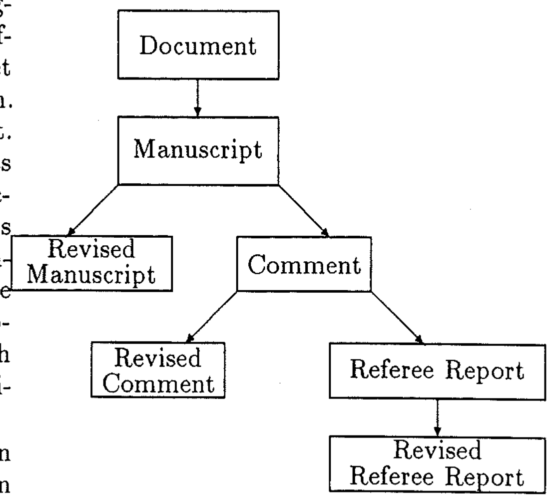


Figure 1: Hierarchy of InterJournal Documents

6 Conclusions

The growth of the Internet, originally motivated by scientific information exchange has not realized its potential in electronic publishing. A new stage of growth of research communication can be achieved if electronic publication can be effectively implemented including various tools for refereeing and article classification to enable selective access by readers. The format and framework of *InterJournal*, while departing significantly from conventional paper journals in various ways, is designed to achieve these objectives and maintain the benefits of peer reviewed publication.

J.UCS and Extensions as Paradigm for Electronic Publishing

Hermann Maurer
(Graz University of Technology, Austria
hmaurer@iicm.tu-graz.ac.at)

Klaus Schmaranz
(Graz University of Technology, Austria
kschmar@iicm.tu-graz.ac.at)

April 21, 1995

Abstract

In this paper we first discuss briefly why electronic publishing up to now had only moderate success. We then describe the philosophy of J.UCS - the Journal of Universal Computer Science - as a possible prototype for electronic publishing in the future. One important facet of the philosophy of J.UCS is the concept of submission and distribution of papers in PostScript as well as in hypertext format. In the following section we discuss the concepts developed to overcome the problems of making PostScript documents hyperlinkable and searchable. The paper concludes with remarks on future extensions of today's electronic journals and the techniques used to distribute papers between mirror sites. Parts of these techniques are already implemented in J.UCS.

1 Introduction

With nearly 4 million internet nodes at the time of writing Internet is the biggest network mankind ever had. Surfing the Internet one will find lots of "electronic paperware" distributed in many different ways. The most elegant way to distribute electronic publications today are distributed hypermedia systems. But until now electronic publishing has been rather unattractive for a number of reasons [see also Maurer, Schmaranz 94].

- Special file formats are used for hypertext. The great variety of modern wordprocessors used today makes it nearly impossible to write filters to convert all the different formats to a specific hypertext format needed. Thus authors are often forced to give up their well known wordprocessing systems and instead deal with completely new and unknown software. Additionally standard hypertext formats (such as HTML or HTF) support a mixture of text and inline images but are lacking symbol charactersets. Hence, in a typical mathematical paper, all the formulae have to be included as inline images causing unnecessary work for the authors and producing large amounts of data.

- Electronic documents are usually without page number information. This makes it impossible to have literature references pointing to an electronic article in a way similar to paper based articles.

- Already existing books and papers are mostly in formats that make it difficult to convert them to hypertext.

- Archival material that has been acquired by scanning or by reprocessing laser printer output is very likely to be in PostScript form rather than in hypertext form.

- The real power of electronic journals is found in the possibility to provide navigational facilities that make it easy to locate interesting articles. Very often those facilities are limited to a simple title search. This is surely not enough.

- Electronic journals today tend to be similar to their paper-based counterparts. They could also contain non-printable information such as animation, 3D models and sound as an explanatory add-on to the text. This would make them more useful in a number of situations.

- Data has to be transmitted over long distances; during rush hours transmission rates are inacceptably low and very often there is only a single server worldwide.

- All large Hypermedia systems such as WWW, Gopher and WAIS are missing built-in billing mechanisms making charging for electronic journals tedious.

The Journal of Universal Computer Science (J.UCS) is a possible prototype of the kind of electronic journal publishing of the future. Section 2 of this paper describes the philosophy of J.UCS that makes electronic publishing more attractive for authors and readers than paper based publishing has ever been. In the following section we explain the issue of submitting papers in PostScript format rather than writing specialized hypertext. The remainder of the paper is then devoted to some further new concepts of multimedia extensions in electronic publishing and to a new philosophy in mirroring and caching documents in J.UCS servers.

2 The Philosophy of J.UCS

2.1 J.UCS - A High Quality Journal

J.UCS is a high quality journal in many senses. First of all each submission is scrutinized by a minimum of three referees and accepted only if it measures up to the standards of prestigious printed journals in computer science.

The editorial board [see J.UCS 94a] consists of over 160 eminent computer scientists all over the world covering all areas of computer science. This prominent editorial board ensures that papers in J.UCS will be considered to be as prestigous as papers in any other reputable refereed journal.

The over 60 "Foundation Servers" [see J.UCS 94b] - the original servers distributing J.UCS are found at many prominent universities and organisations worldwide.

The reputation of J.UCS as a high quality journal does not only depend on the quality of the published papers but also depends on its stability. In some other electronic journals papers once published change from time to time as new research results get known. J.UCS is stable in the sense that no article once appearing will ever be changed at a later stage with the exception of annotations that can be added. This is very essential to be able to quote contributions without fear that they can change or even disappear.

Annotations in J.UCS are used to alert the reader of errors and new research results and are implemented using hyperlinks. Annotations in J.UCS are also refereed exactly like papers to ensure their correctness and to avoid their use for personal disputes.

Publications in J.UCS are structured into pages that are numbered consecutively so that papers can be quoted exactly like in usual journals with name(s) of author(s), title, name of the journal, volume number, issue number and page number(s).

Stability of J.UCS is also essential because J.UCS does not only appear in electronic form. Springer will also provide a yearly CD-ROM version and a yearly printed version for archival purposes. The CD-ROM version will have the same hyperlink, multimedia and print facilities as the electronic version distributed in the Internet.

With the permission of ACM papers appearing in J.UCS are categorized following the ACM Computing Reviews categories. A complete overview of the possible categories is given in every January edition of ACM Computing Reviews [see also ACM 94] as well as in J.UCS [see J.UCS 94c].

2.2 J.UCS - Universal in Many Senses

J.UCS is designed to be universal in many senses: First of all it covers all aspects of computer science. All known paper based journals only cope with a part of the wide variety of areas of modern computer science because they would turn into paper monsters that are impossible to read [see Odlyzko 94]. Due to its electronic nature this is not a problem for J.UCS. The highly sophisticated search methods of Hyper-G [see also Andrews, Kappe 94] - the kernel of J.UCS - allow it to locate interesting papers easily.

The second aspect of universality is its distribution in the Internet: readers can access J.UCS at any time, day and night and at any place worldwide. Using Hyper-G servers for J.UCS distribution makes J.UCS again more universal for readers: They can use one of the native Hyper-G clients Harmony or Amadeus or they can also read J.UCS with one of the well known WWW or Gopher clients. Since Hyper-G has many features that are not supported by WWW or Gopher the use of those clients causes some loss of functionality but readers of J.UCS are not forced to leave their well known environment.

J.UCS is distributed in two parallel formats: hypertext and PostScript. Again being universal both formats provide the same functionality. With Hyper-G it is possible to have hyperlinks in PostScript files. Also a full text search engine for PostScript documents is provided. The hypertext version of papers contains page number information and it is divided into abstract and single sections to allow the reader quick browsing. The reader can then decide to read the paper on screen or to get the PostScript version for high quality printing.

Universal also means that Internet is not the only way of distribution - readers that have no access to the Internet can get a CD-ROM or even a paper based version by Springer. The CD-ROM version provides

the full functionality known from the Internet version. Naturally, page numbers are the same in each of the three versions.

¿From our point of view universal access does not only mean getting papers but getting them fast. For this purpose a world wide net of over 60 servers guarantees short distance access and high data transmission rates. Additionally one server need not deal with all readers simultaneously, hence reducing response time. J.UCS issues are transfered to the servers as they appear and are considered static in the sense that they do never change with the exception of annotations.

Quick access not only means high transmission rates but also includes sophisticated methods for locating interesting papers. Therefore J.UCS provides powerful facilities to search for keywords in the title, in the list of keywords supplied by the author or even in the whole text, by author, by category, by date or by combinations thereof. As an example, searching for all papers between 95 and 96 with classification H.3 or "Information Storage and Retrieval" will produce a "subjournal" of all papers of J.UCS published in those two years and classified as contributions to "Information Storage and Retrieval". Note that contributions need not be necessarily classified under only one category. Consider a typical paper on "Hypertext" - this might be classified as H.3 ("Information Storage and Retrieval"), H.4 ("Informations Systems Applications") and I.7 ("Text Processing").

Universality is not only understood as being universal from the reader's point of view - J.UCS is also universal from the author's point of view. The main submission format is PostScript. Nearly every wordprocessor today is able to produce PostScript output, so authors of J.UCS papers are not forced to leave their well known word processing environment when submitting a paper to J.UCS.

Last but not least J.UCS is also universal from the publishing company's point of view. Publishing electronically in the Internet does not necessarily mean that the published journal must be free of charge. For a trial period of 2 years between 1995 - 1996 J.UCS is freely available. After this trial period it is necessary to collect charges to recover the central server and network costs.

For this purpose a billing mechanism for Hyper-G was implemented allowing to keep track of simultaneous users of a certain issue of J.UCS. Thus organizations can manage the access to J.UCS issues just as is the case in libraries: the organization pays fees for a specified number of J.UCS versions and access to one issue of J.UCS is then limited to this specified number of simultaneous readers. J.UCS, although not intended to be a free publication will certainly be less expensive than comparable printed journals. As a result of the electronic nature of J.UCS all costs of printing and mailing will disappear.

3 PostScript - Better Than Hypertext

The use of specialized hypertext formats has some disadvantages for authors, readers and also for information providers:

- From the author's point of view hypertext formats are not general enough and poorly supported by today's wordprocessing systems. Hypertext formats are document-based rather than page based which makes quoting of other hypertext papers difficult. The author has no possibility to include layout information in the document because all hypertext viewers are reformatting the document according to their window size.

- Readers have to deal with documents that are drafts rather than high quality printable papers. Additionally special characters such as formulae have to be included as inline images making hypertext papers bigger than their PostScript counterparts.

- Information providers such as publishing companies often want to publish reprocessed archive material. This material is very likely to be either in PostScript or in some other page based layout description rather than in hypertext.

For this reasons a file format had to be chosen meeting the following requirements:

- **Compatibility:** The format has to be supported by most wordprocessing systems today. Additionally it must be possible to easily convert existing archival material to that format.

- **Flexibility:** There must be no restrictions in charactersets and layout information to allow full featured documents and high quality printing.

- **Convertability:** The format must be convertible to traditional hypertext formats to provide backward compatibility to today's hypermedia clients.

- **Linkability:** Hyperlinks must be supported to make documents usable in hypermedia systems.

- **Searchability:** Full text searches must be possible for use in hypermedia systems.

Considering the requirements the points compatibility and flexibility are most difficult to fulfil. One format providing the demanded features is PostScript. There is also another format that is as flexible as PostScript: *pdf* see also [Adobe 93] and [Brailsford 94]. Additionally *pdf* has some advantages compared to PostScript (e.g. it has a page-index), but at the moment it is not as widespread as PostScript. PostScript is fully supported by today's software. Please note that the concepts proposed in this paper will also work with *pdf* and *pdf* will be supported in J.UCS if *pdf* becomes sufficiently widespread.

Nearly every wordprocessing system today is able to produce PostScript output, at least using a PostScript printer interface and redirecting the output to a file. Existing archival material is very likely to be already in PostScript form or to be in a format that can be easily converted to PostScript.

But what about Convertability, Linkability and Searchability? PostScript is a page based layout language. PostScript documents are not guaranteed to contain text in the form of useful textual information to allow searching or conversion; nor does PostScript support hyperlinks.

Hyperlinks can be easily added to PostScript documents when a hypermedia system uses a link database as is the case with Hyper-G. Hyper-G servers support PostScript documents with links. The native Unix Hyper-G client Harmony [see Andrews, Kappe 94] and Amadeus already have a PostScript Viewer for hyperlinked PostScript documents. Source as well as destination anchors are simply defined by page number and coordinates and inserted in the link database. Due to the definition by coordinates anchors can appear anywhere on a page marking words, paragraphs, figures or parts of them and they can be partially or completely overlapping.

As one can see using Hyper-G for distribution of "electronic paperware" is the solution for nearly all problems of today's electronic publishing including hyperlinked PostScript documents. The last remaining problems are making PostScript searchable and converting PostScript to hypertext formats for backward compatibility.

As it turns out there are techniques to get text and images out of PostScript documents. With that information one can build a full text index and one can also convert PostScript to other formats. Since Hyper-G supports full text indexing also for non-text documents it is again Hyper-G that comes in handy for solving that problem.

4 Future Extensions to Electronic Journals

One problem of today's electronic journals is the fact that they are often too similar to their paper based counterparts. Mostly they contain only printable information such as text and images. Utilizing the full power lying in the nature of electronic journals allows to go much further.

4.1 Multimedia Add-ons

The first step to future electronic publishing is to allow multimedia add-ons to papers. They are stored as appendices seperately in the Hyper-G database and they are represented by a hyperlink in the paper making them available with a simple mouseclick.

Both the online version and the yearly CD ROM version of J.UCS fully support printable and non-printable multimedia add-ons such as images, sound, movies and even 3D scenes.

Very often, especially for educational purposes, it is also useful to provide guided tours as explanatory add-ons.

The yearly printed version of J.UCS contains textual information and printable add-ons such as images. For non-printable add-ons such ad 3D scenes and movies the authors have the possibility to make snap shots of certain situations that can be printed as images or series of images.

For future extensions the term multimedia is not only limited to sound, movies and 3D scenes. Multimedia can also mean much more sophisticated and specialized document types than the ones mentioned before:

- Specialized viewers for mathematical expressions allowing to evaluate formulae, calculate series of curves, plotting surfaces of graphs and much more.

- Specialized viewers for algorithms allowing to execute pseudo code to get an exact idea how an algorithm works.

- Specialized viewers for other groups of interest such as molecular structure viewers for research in chemistry.

- Guided tours for educational purposes as well as question-answer dialogs. The guided tours themselves can contain either document type mentioned in this chapter.

4.2 Dictionaries

As science becomes more and more specialized there are many new technical terms and it becomes impossible for readers to know all of them. For this purpose we will implement special dictionaries in Hypermedia systems where the reader can quickly look up unknown terms.

For large hypermedia systems the dictionary has to be organized by "input" and "output" topics to speed up the search. For example "input" topics could be computer science, mathematics, physics etc. and "output" topics could be explanations of the terms or translation of the terms to a particular language. The range of topics that are considered in a lookup operation is controlled by special attributes of a specific document and by user preferences.

For example native English speakers searching for the term hyperlink in a paper on hypermedia systems will get an English explanation for the word hyperlink because their prefered search attribute is "english explanation" and the search will be performed in the dictionary subset "computer science" because of the attribute of the paper.

4.3 Reviews

The nature of electronic journals allows easy insertion of new documents to hypermedia databases. One interesting feature of this is to provide the possibility for readers to insert (refereed) reviews of articles in the database. Compared to the full papers the reviews are much smaller. If readers want to browse a database to find interesting articles they can simply fetch the reviews and then decide which article they want to get.

Because the reviews are much smaller than the papers they can easily be mirrored across servers. So readers can browse a server geographically convenient for them and then only need to download papers over long distances if the paper suits their interests.

4.4 Limited Access

In many cases electronic journals or single papers have to be accessible by only a limited user group. In the case of journals this may e.g. be only the subscribers of the journal or in the case of conference proceedings only the registered conference participants.

One way to limit access is to manage user and group accounts. However conference participants having an account could give away their password so that many other people would also have access to the journal or proceedings without paying any fees. Hence accounting without further mechanisms is certainly not enough for such cases.

A better way is to manage accounts together with the number of downloads or the address of the computer from which the reader is allowed to view the papers. If users are allowed to download a single paper at most three times they certainly will not give away their password. The same is the case with the address of the downloaders: if papers can only be downloaded from a registered computer there is no sense having the password and trying to get the paper from a not registered computer.

5 Server Philosophy

This last chapter in this paper deals with some techniques that make the life of server sites and information providers easier as well as the life for readers of electronic journals.

One big problem why electronic journals today are considered to be less attractive than their paper based counterparts is their availability. In many cases there is only one single server world wide forcing long distance data transfers for most of the readers. During rush hours the transfer rate over the Internet is inacceptably low. A second disadvantage of distribution of journals with a single server is the machine load: the more readers are logged in, the higher the machine load gets and the more sluggish the system responds.

One way out of this dilemma is to have mirror sites all over the world as is the case with J.UCS. Readers can choose a server geographically convenient for them. Then data transmission speeds remain acceptable even during rush hours. The second effect of mirroring is that the number of readers per server significantly decreases and the system load is kept low.

But this concept has also some disadvantages. First, the servers have limited disc space and some organisations do not want to mirror the whole part of the database needed for the journal. Second, data has to be transmitted periodically from the main server to all mirror servers in one piece.

Those problems can be solved by a concept we call "super-caching". This concept is based on the idea of caching with some additional features.

The only data that are transmitted in one block are

references to new papers. So every information that is needed to access a single paper is guaranteed to be available on every mirror site. The references to the papers also contain some some additional attributes that are evaluated by the cache of the mirror site.

Additional attributes describe the nature of the paper such as: *review*, *hypertext paper*, *PostScript paper*, *movie*, *3D scene*, etc. The server operator can than decide how to deal with the special types of papers. Possibilities are to let the server automatically mirror papers or only mirror them on demand but then keep them in the database or simply download them on demand and keep them in the cache until they are falling out of the cache automatically.

As an example a big mirror site can be set up to automatically mirror all data and keep them in its database. A medium sized mirror site can download reviews and papers automatically and multimedia add-ons are only downloaded on demand. A small mirror site could even only download the reviews and abstracts of papers to allow quick searches for papers in a browsing mode; the reader gets full papers and add-ons on demand only.

This concept of distributing data between the main server and mirror sites can even be implemented better if a pyramid of distribution servers is built up. The main server only transfers the references to papers to its nearest mirror sites. Those mirror sites themselves are masters for some other mirror sites and transmit the references to them and so on.

In this case it is necessary to transmit a little bit more than the plain references. The nearest location of the papers has also to be transmitted because there can be some mirror sites in the pyramid that do not mirror all the data but other servers below them do.

The algorithm for distribution of this list is easy: The first server builds the list and inserts its address for each paper. This list is transmitted to the next level of the pyramid. Each server in this level replaces the address of the first server by its own for each paper it mirrors automatically. The rest of the addresses remains unchanged. Then the server transmits the updated list to the next servers. They themselves mirror the desired papers from the server given in the list and replace those addresses by their own and so on until the end of the pyramid is reached.

References

[ACM 94] "The Full Computing Reviews Classification System"; Computing Reviews 35, 1 (1994), 6-16.

[Adobe 93] "Portable Document Format Reference Manual", Adobe Systems Incorporated, Addison Wesley (1993).

[Andrews, Kappe 94] "Soaring Through Hyperspace: A Snapshot of Hyper-G and its Harmony Client"; Proc. of Eurographics Symposium on Multimedia/Hypermedia in Open Distributed Environments, Graz, (1994), 181-191.

[Brailsford 94] "Experience With the Use of Acrobat in the CAJUN Publishing Project", Proceedings ECHT'94 (1994).

[J.UCS 94a] "J.UCS Editorial Board", http://www.iicm.tu-graz.ac.at/Cjucs_general_editorial_board

[J.UCS 94b] "About J.UCS", http://www.iicm.tu-graz.ac.at/Cabout_JUCS

[J.UCS 94c] "The Full ACM Computing Reviews Classification System", http://www.iicm.tu-graz.ac.at/Cjucs_general_categories

[Maurer, Schmaranz 94] aurer H. and Schmaranz K., "J.UCS - The Next Generation in Electronic Journal Publishing", Proceedings NSC'94 and Computer Networks and ISDN Systems, Computer Networks for Research in Europe vol. 26 Suppl. 2,3, (1994), 63-69.

[Odlyzko 94] Odlyzko, A., M.: "Tragic Loss or Good Riddance? The impending demise of traditional scholarly journals"; to be published in "Electronic Publishing Confronts Academia: The Agenda for the Year 2000," Robin P. Peek and Gregory B. Newby, eds., MIT Press/ASIS monograph, MIT Press, (1995)

The SIGACT Theoretical Computer Science Genealogy: Preliminary Report

Ian Parberry*
Department of Computer Sciences
University of North Texas

David S. Johnson†
AT&T Bell Laboratories

Abstract

The SIGACT Theoretical Computer Science Genealogy, which lists information on earned doctoral degrees of theoretical computer scientists, is currently in the process of being published on the World-Wide Web. We describe the document, its applications, and some simple statistics.

1 Introduction

The SIGACT[1] Theoretical Computer Science Genealogy lists information on earned doctoral degrees (thesis adviser, university, and year) of theoretical computer scientists worldwide. The genealogy was initially published in print form over a decade ago, and included a listing of the entire genealogy (Johnson [1]). However, the genealogy has since doubled from 554 entries listing 665 names to 1196 entries listing 1369 names, making it impractical to print the entire genealogy in the archival literature. Instead, the genealogy will be published electronically over the World-Wide Web as a collection of `html` files. A preliminary version is currently available [6]. An added bonus is that it is now possible to explore the intellectual ancestry of individuals in the genealogy by following a series of hypertext pointers.

The Theoretical Computer Science Genealogy is intended as an informational tool. Its main application is undoubtedly entertainment, but it does have more formal uses. At various times in the past, Program Directors at the National Science Foundation have used the genealogy to avoid possible conflicts of interest caused by having a funding proposal refereed by the doctoral adviser or student of the investigator. We envisage editors of refereed journals using it for the same purpose.

The remainder of this document is divided into four sections. The first describes the organization of the World-Wide Web version of the TCS Genealogy. The second describes the text database from which the `html` files are generated. The third describes some simple statistics about the TCS genealogy that can easily be obtained from the `html` files. The fourth describes the work remaining to be done before the genealogy is ready for a full release.

2 Organization

The World-Wide Web version of the TCS Genealogy is divided into a large number of files so that users who browse only a fraction of the genealogy will not have to wait while large amounts of unnecessary data are transferred across the Internet. The overall structure of the genealogy is shown in Figure 1 (many of the links are condensed or omitted to enhance readability). The major parts of the genealogy are the main index, the submission details page, the online form, the text file page, the statistics page, the name index, the letter indices, the university index, the country indices, the university indices, the year index, the decade indices, and the main database. Each of these is described briefly in a separate subsection below.

2.1 The Main Index

The main index is the first thing that the user sees, and is therefore very brief. It contains links to the SIGACT page [3], the submission details page, the text file page, the statistics page, the name index, the university index, and the year index (see Figure 2).

*Author's address: Department of Computer Sciences, University of North Texas, P.O. Box 13886, Denton, TX 76203-3886, U.S.A. Electronic mail: `ian@ponder.csci.unt.edu`. URL: `http://hercule.csci.unt.edu/ian`.

†Author's address: AT&T Bell Laboratories, 600 Mountain Avenue Rm. 2D-150, Murray Hill, NJ 07974, U.S.A. Electronic mail: `dsj@research.att.com`.

[1] SIGACT is the acronym for the ACM Special Interest Group on Algorithms and Computation Theory. More information about SIGACT is available on the World-Wide Web [3].

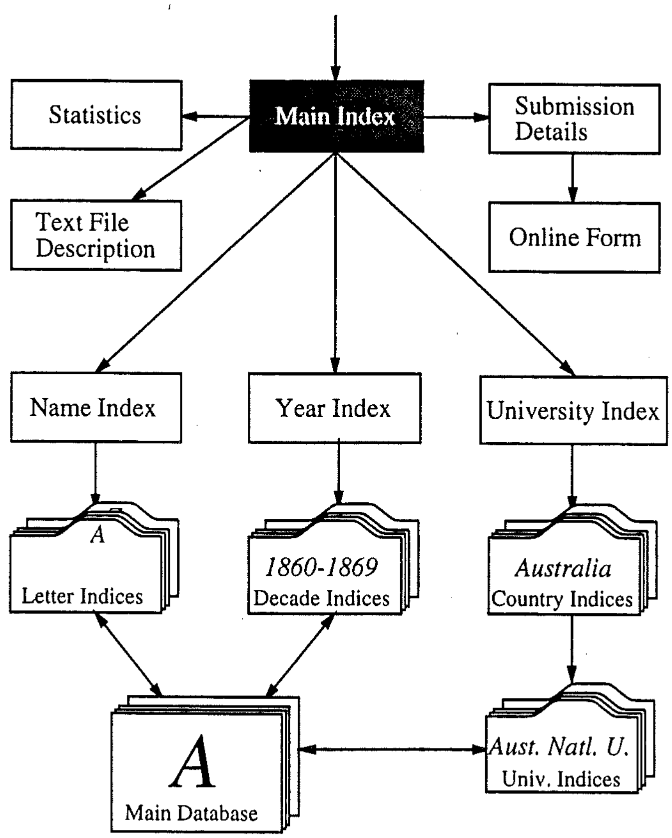


Figure 1: Flowchart showing main `html` files and the primary links.

The Theoretical Computer Science Genealogy

Welcome to the SIGACT Theoretical Computer Science Genealogy, which lists information on earned doctoral degrees (adviser, university, and year) of theoretical computer scientists worldwide. More information about submission details and entry criteria is available. The TCS Genealogy is also available as a text file. Some interesting facts about the TCS Genealogy are also available.

Entries in the TCS Genealogy are indexed by:

name,
university, and
year.

This is a pre-release version of the genealogy, which may contain some bugs.

Created by Ian Parberry, October 9, 1994.
Last updated Tue Dec 20 10:06:25 CST 1994.

Figure 2: The main index. Underlining indicates hypertext links.

2.2 The Submission Details Page

The submission details page contains information on how to submit an update, what information is needed in an update, and what qualifications are necessary for entry into the genealogy. Basically, a person must have made a contribution of some kind to theoretical computer science, loosely defined as at least one of the following:

1. an article published in refereed theoretical computer science journal,
2. a conference paper in a leading theoretical computer science conference,
3. regular attendance at a leading theoretical computer science conference,
4. being sufficiently famous that most readers will recognize one, or
5. an ancestor of an existing entry.

Except for people qualifying under (5), one must have officially received one's PhD before one can be entered into the database.

The submission details page also provides access to the online form.

2.3 The Online Form

The online form lets users submit entries to the genealogy using browsers that support fill-out forms. Figure 3 shows a screen shot of the top of the form using NCSA Mosaic. Before the World-Wide Web version of the genealogy was conceived, entries were submitted by sending email to `pedigree@hercule.csci.unt.edu`. For consistency, the online form automatically emails completed forms to the same address. Updates are not fully automatic, however. Each entry must be processed by hand to ensure consistency (for example, Richard Karp has been referred to in various updates as R. Karp, R. M. Karp, Richard M. Karp, and Dick Karp) and perform error-checking (for example, spelling, and checking that the fields were entered in the correct order).

2.4 The Text File Page

The text file page explains the format of the text version of the genealogy, and allows `ftp` access to the text files.

2.5 The Statistics Page

The statistics page lists a few very simple statistics about the genealogy that were gathered automatically.

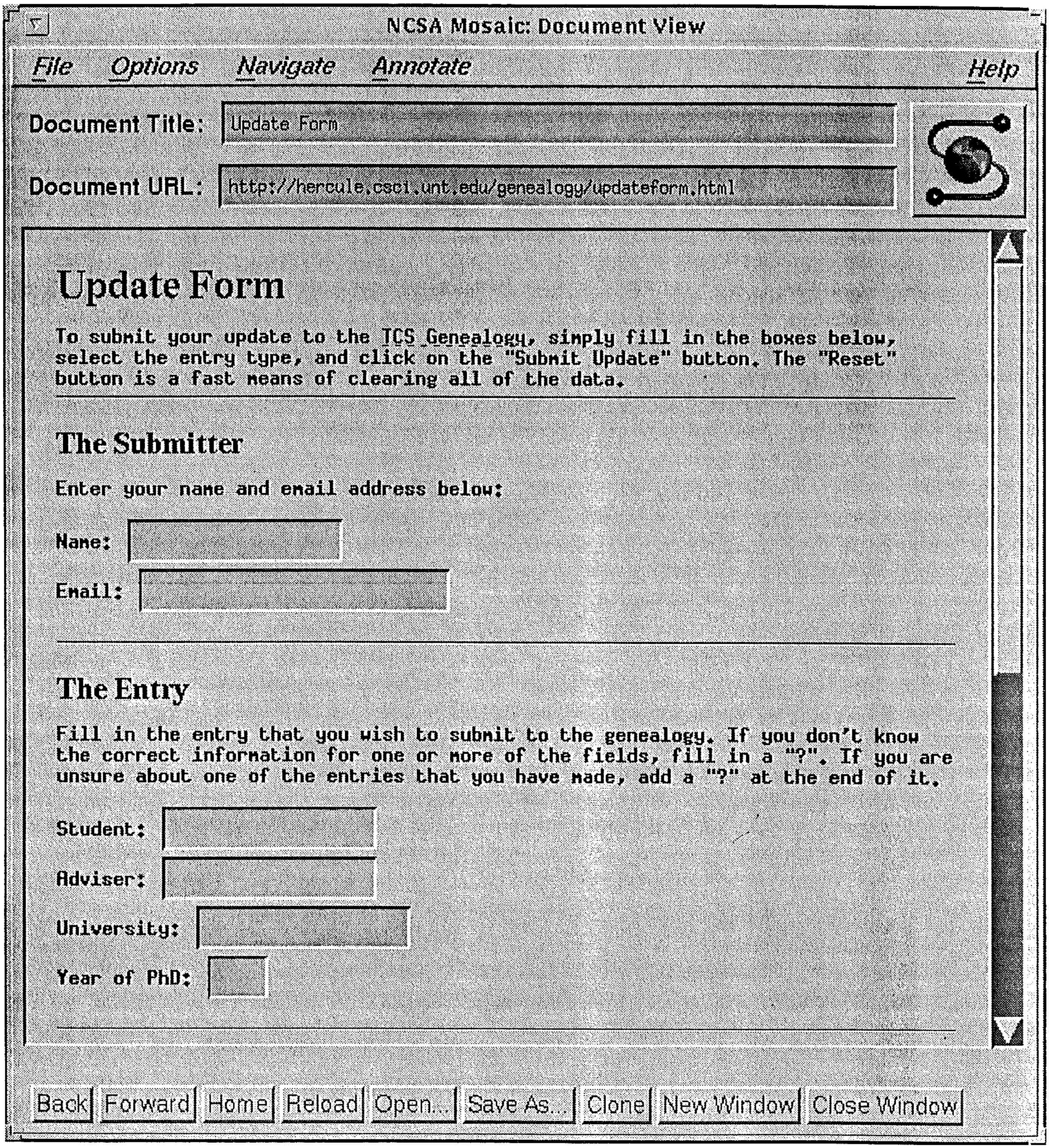


Figure 3: Screen shot of the fill-out form using NCSA Mosaic for X windows.

2.6 The Name Index

The name index contains hypertext links to the letter index files (see Figure 4).

2.7 The Letter Indices

There is a letter index file for each letter of the alphabet. The letter index file for the letter "A", for example, contains a hypertext link to the main database entry for each person whose last name begins with the letter "A".

2.8 The University Index

The university index allows access to the main database according to the the university that granted the doctoral degree. It contains hypertext links to the country indices.

2.9 The Country Indices

There is a country index for each country mentioned in the genealogy. Each country index contains hypertext links to the university indices for the universities in that country.

2.10 The University Indices

There is a university index for each university mentioned in the genealogy. Each university index gives the full name and geographic location of a university, and hypertext links to the main database entries of its doctoral graduates.

2.11 The Year Index

The year index allows access to the main database by year of graduation. It contains hypertext links to the decade indices.

2.12 The Decade Indices

There is a decade index for each decade mentioned in the genealogy. Each decade index has a section for each year in the corresponding decade. Each year section contains hypertext links to the main database entries of doctoral candidates who graduated in that year.

2.13 The Main Database

The main database consists of 26 `html` files, one for each letter of the alphabet. The database file for the letter "A", for example, contains the entry for each person whose last name begins with the letter "A". Each entry lists the person's name, the university from which they received their doctorate, and the year in which the degree was granted, followed by a list of their doctoral students, and the universities and years in which their doctoral degrees were granted. Each of these pieces of information is a cross-reference to information in another part of the database.

For example, Figure 5 shows the entry for the first author of this paper. The first line lists his name. The second line states that he obtained his degree from Warwick University in 1984. The text "Warwick University" is a hypertext link to the index for Warwick University, and the text "1984" is a hypertext link to the index for the year 1984. The third line states that his adviser is Mike Paterson. The text "Mike Paterson" is a hypertext link for the main database entry for Mike Paterson (where the browser will see Ian Parberry listed as one of his students). The succeeding lines list Ian Parberry's doctoral students, with hypertext links to their main database entries, and to the indices for the university and year of their respective doctoral degrees. The last line contains a link to the submission details page.

3 The Text Database

The text version of the database consists of two files, the database file, and the university file. Each is described below in a separate subsection. The text files are the canonical version of the genealogy. The hypertext version of the TCS Genealogy is created automatically from the text files by a Unix shell script (using `sed` and `grep`) written by the first author.

3.1 The Database File

The database file contains the main database. It consists of a header, followed by the entries. Each line of the header begins with the character "#". Each entry consists of four fields separated by a single tab character. The fields are, from left to right:

1. the student's name,
2. the name of the student's thesis adviser,
3. an acronym for the university granting the doctoral degree (see below), and
4. the year the degree was granted.

A student with multiple doctoral degrees has one entry for each. A student with multiple advisers for a single doctoral degree also has multiple entries (one for each adviser), but the university and year are the same.

A field consisting solely of the character "?" indicates that the information in that field is unknown. The

"?" character is also used to indicate that the information provided in a field may be incorrect. An entry for a person without a doctoral degree (which is included when he or she has served as a thesis adviser on doctoral degrees) has the string "---" (three hyphens) in the adviser, university, and year fields.

3.2 The University File

The university file maps acronyms to universities. Each entry consists of an acronym, followed by the character "=", followed by the name, city or town, state or province (if applicable), and country of a university (separated by commas).

4 Statistics

Since the database is maintained electronically, it is relatively easy to gather some simple statistics. The remainder of this section is divided into two subsections. The first contains statistics about the database files, and the second contains statistics about the TCS Genealogy itself. The information reflects the state of the genealogy as of December 20, 1994.

Note that statistics from the TCS Genealogy do not necessarily reflect the whole of the theoretical computer science community. Much of the information in the original database was obtained by personal solicitation from the second author (in person or via email), and despite his intent to be as universal as possible, the information he obtained probably reflects at least a slight bias toward those areas (both geographic and technical) with which he was most familiar, as well as the school (MIT) that he attended. Subsequent entries are biased in different ways. So far they have for the most part been obtained as a result of general solicitations, rather than individual arm-twisting, and so people who do not normally read or respond to such solicitations have a higher probability of being absent. We hope to rectify this in the near future.

4.1 The Database Files

The genealogy consists of 240 html files, which are cross-referenced using a total of 11126 hypertext links (HREFs), and take up a total of 1.185 MB of file space.

Country	Count
Australia	1
Austria	3
Belgium	1
Bulgaria	1
Canada	6
Denmark	1
England	6
Finland	3
France	3
Germany	18
Hungary	2
Israel	5
Italy	2
Japan	1
Norway	1
Poland	3
Prussia	1
Russia	5
Scotland	1
Spain	2
Sweden	2
Switzerland	2
The Netherlands	4
USA	67

Table 1: Number of universities mentioned in the TCS Genealogy by country.

4.2 The Data

The TCS genealogy contains entries for 1369 scientists with last names starting with 24 of the 26 letters of the alphabet (the exceptions are "Q" and "X" — we may be able eventually to get up to 26 letters, since there are three authors whose names begin with "X" and one whose name begins with "Q" in the STOC/FOCS bibliography [2]). The most frequent letter is "S", with 179 entries. A frequency graph is shown in Figure 6.

University	Count
Columbia University	10
Edinburgh University	10
University of Maryland	10
UCLA	10
Brown University	11
Georg-August-Universitat Gottingen	11
University of Michigan	11
Warsaw University	11
Purdue University	12
University of Turku	12
University of Southern California	12
Yale University	12
Utrecht University	13
University of Chicago	15
University of Minnesota	16
Hebrew University	19
Weizmann Institute	20
University of Wisconsin	20
University of Waterloo	21
Penn State University	25
University of Toronto	25
University of Washington	25
University of Illinois at Urbana-Champaign	35
Carnegie Mellon University	36
Harvard University	55
Stanford University	68
Cornell University	69
Princeton University	70
University of California at Berkeley	77
MIT	94

Table 2: Number of entries from universities that have at least ten entries.

The genealogy contains entries from 141 universities in 24 countries. Most entries are from the US (see Table 1). A total of 30 universities have at least 10 entries (see Table 2). As expected, MIT has more entries than any other university.

The number of entries in each decade grows rapidly from the 1940s through the 1970s (see Figure 7). The entries before the 1950s are mainly ancestors of theoretical computer scientists. A closer examination of the data since 1960 (see Figure 8) reveals that the number of entries per year has roughly leveled out since the early 1970s.

5 Remaining Work

A small amount of work remains to be completed before the WWW genealogy is ready for full public release. Some things are currently done incorrectly, including the following.

- There is no distinction between official advisers, unofficial advisers, and co-advisers.
- Dual doctorates are not handled properly (the genealogy currently contains two dual doctorates, Andrew Yao and Leonid Levin).
- Accents in foreign names are omitted.
- Compound last names (such as Meyer auf der Heide, and van Emde Boas) are not alphabetized correctly.

Until then, a pre-release version is available [6]. Please feel free to browse it and report any errors, bugs, or updates to the first author.

Some additional features to be added at a later date include:

- Create one `html` file for each person, rather than one for each letter of the alphabet. This will decrease downloading time substantially.
- Add links to the home pages of people who have them. A list of such links is already available in the TCS Virtual Rolodex [5]. All that remains is to integrate them with the genealogy.
- Allow the inclusion of small pictures of each individual in the genealogy.
- The student-supervisor relationships form a DAG. Provide the ability to do online queries on the DAG, including properties such as connected components, paths, cycles, least common ancestors, and graph drawing.

The final version of this report, to be published in *SIGACT News* (see [4]), will include more information on the database, including issues that were covered in the original report [1] such as directed and undirected cycles, and connected components. This information will be computed automatically from the main database. We also plan to develop methods for drawing "family trees" in postscript format. Finally, as mentioned in Section 4, the authors plan to start soliciting genealogical information from individual members in the theoretical computer science community, starting with names mentioned in the STOC/FOCS bibliography [2], and attendee lists from recent theory conferences.

References

[1] D. S. Johnson. The genealogy of theoretical computer science. *SIGACT News*, 16(2):36–44, 1984. Reprinted in *Bulletin of the EATCS*, (25):198–211, 1985.

[2] D. S. Johnson (Editor). STOC/FOCS Bibliography (Preliminary Version). ACM Press, 1991.

[3] I. Parberry. ACM SIGACT. A WWW document with URL `http://sigact.acm.org/sigact`, 1994.

[4] I. Parberry. SIGACT News. A WWW document with URL `http://sigact.acm.org/sigactnews`, 1994.

[5] I. Parberry. The Theoretical Computer Science Virtual Rolodex. A WWW document with URL `http://sigact.acm.org/tcs-rolodex`, 1995.

[6] I. Parberry. The Theoretical Computer Science Genealogy. A WWW document with URL `http://sigact.acm.org/genealogy`, 1994.

Name Index

This is the name index for entries in the TCS Genealogy.

Aanderaa to Azar (38 entries)
Babai to Butler (112 entries)
Cadiou to Cutland (80 entries)
Dalen to Dymond (41 entries)
Earley to Even (18 entries)
Fagin to Furst (49 entries)
Gabbay to Gusfield (97 entries)
Haber to Huynh (78 entries)
Ibarra to Iwasawa (12 entries)
Ja'Ja' to Joung (21 entries)
Kac to Kutylowski (102 entries)
LaPaugh to Lyuu (86 entries)
Maak to Mylopoulos (103 entries)
Naor to Nodine (17 entries)
O'Donnell to Owicki (20 entries)
Pacholski to Purdom (63 entries)
Rabani to Ruzzo (77 entries)
Sacerdote to Szymanski (179 entries)
Tagamlitzki to Tzeng (56 entries)
Ukkonen to Uspenskij (6 entries)
Vacca to Vuillemin (25 entries)
Waarts to Wyshoff (61 entries)
Yacobi to Yung (18 entries)
Zadeh to Zwick (10 entries)

Created by Ian Parberry, December 13, 1994.
Last updated Tue Dec 20 10:06:57 CST 1994.

Figure 4: The name index. Underlining indicates hypertext links.

> **Ian Parberry**
>
> Doctorate from Warwick University in 1984
> Adviser: Mike Paterson
> Students:
>
> 1. Zoran Obradovic (Penn State University, 1991)
> 2. Bruce Parker (Penn State University, 1988)
> 3. Pei-Yuan Yan (Penn State University, 1989)
>
> Can you help us to update or correct this entry?

Figure 5: The main database entry for Ian Parberry. Underlining indicates hypertext links.

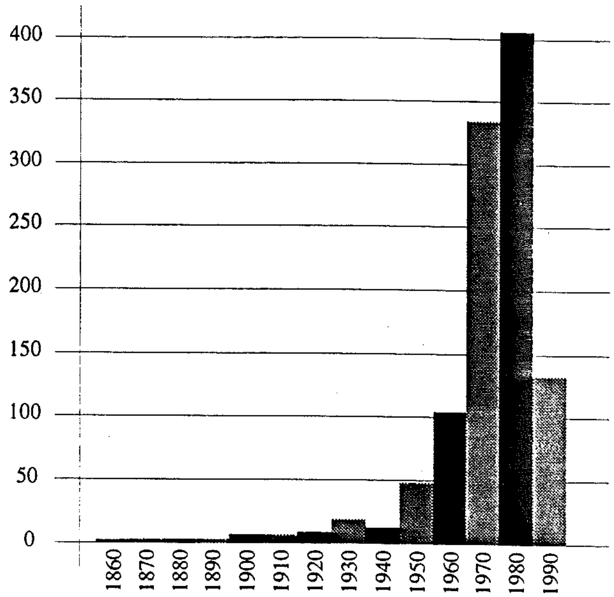


Figure 7: Number of entries in the TCS Genealogy graduating in each decade.

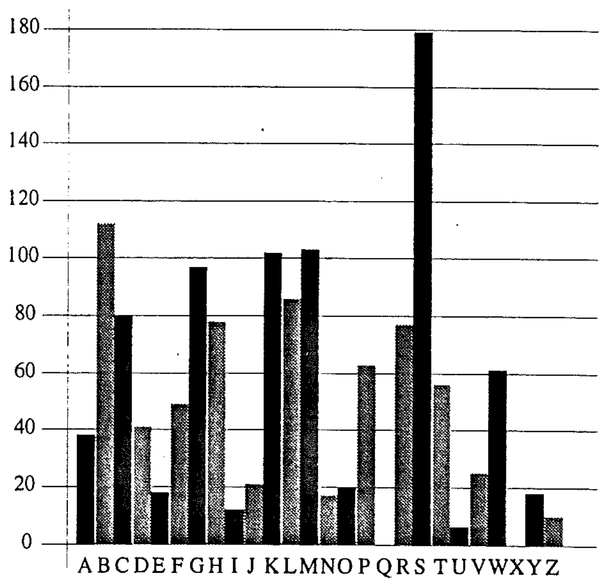


Figure 6: Number of names in the TCS Genealogy starting with each letter of the alphabet.

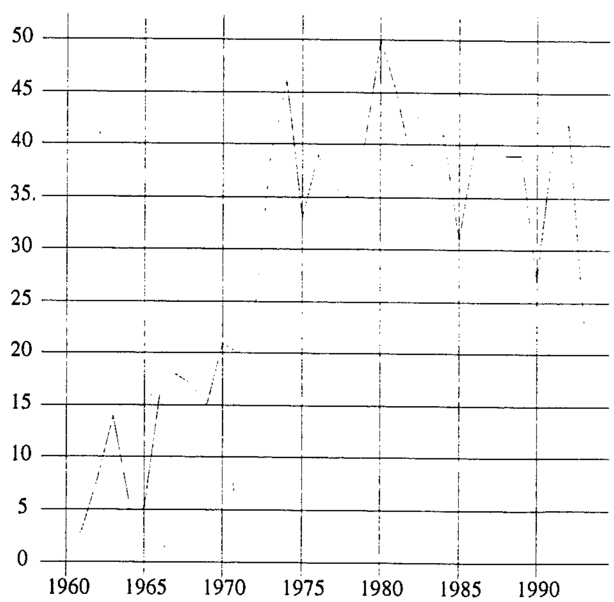


Figure 8: Number of entries in the TCS Genealogy graduating in each year from 1960.

Calculus Modules On-line: An Internet Multimedia Application

by Leslie Bondaryk
Manager of Technology Development
PWS Publishing Company, 20 Park Plaza, Boston, MA 02116

Abstract

Calculus instruction can benefit from interactive computer tools in helping students to visualize, calculate and connect elements in the course. The Web, through the Mosaic interface, provides a hyperlink engine by which a front-end interface and a back-end help system can be created for Computer Algebra System examples and problem files, allowing the course material to be modularized and extensively cross referenced. The system proposed meets many important requirements for an interactive calculus tutoring system, and is an attractive environment from the publisher's perspective, since it allows ready updating of individual files, easy customization of course structure and content, and integration with the goals of the reform movements in college education.

Introduction: interactive calculus instruction

Calculus, as well as other mathematics and science disciplines, has a reputation for being difficult to learn and explain. The reform movement in calculus instruction[1] has attempted to overcome some of the math-anxiety using a variety of approaches, one of which is the incorporation of technology in the classroom[2]. Multimedia programs can provide examples and contextual links in calculus in a way that is clearer and more compelling than can be achieved on paper: animation, sound, and other visualization tools can make complex concepts more accessible, software can be made to perform rote tasks, such as repeated calculations and graphing, and hyperlinking can help a student to understand connections within the course in a way which is more potent than on paper or on a blackboard.

PWS Publishing wants to create a calculus tutoring environment which takes some of the mystery and drudgery out of the calculus course, allowing the student to concentrate on the course concepts, and perhaps get more excited about mathematics in general. Such a system should also support our printed Calculus textbooks Additionally, we wanted a system which is more easily updated or reconfigured than a printed text; this is particularly important in core college courses like Calculus, since the teaching methods and materials used in this class are frequently updated, and since the calculus curriculum is often custom-structured to suit the particular needs of students and teachers. Presently, many calculus instructors would even like to be able to include their own materials as part of professionally published texts, a practice known as "custom publishing."

Project Description

The first kind of courseware published by many publishing houses were simply files which could be used in other commercial software systems. It has become common practice in much of the calculus teaching community to use Computer Algebra Systems (CAS) such as Maple, Mathematica, Mathcad, etc. to teach mathematics. Often, disks of CAS files which demonstrate concepts in a textbook would be bound into the back of the book, or could be purchased to separately[3,4]. Sets of CAS files do provide a good set of templates for students to use in solving homework and laboratory problems in calculus, but they don't fully exploit the promise of multimedia educational tools.

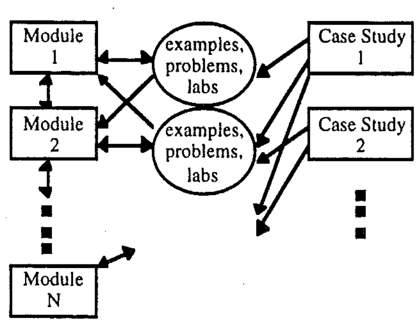


Figure 1

We have designed a system that meets more of the goals of calculus reform and the publishing industry, and are prototyping it on the World Wide Web[5], using the Mosaic[6] interface. The general structure is

shown in Figure 1. The best way to achieve the level of customization and interactivity required for this project was to create a modular system which would center the course materials not around chapters, but rather around curriculum segments: the Derivative, the Integral, Functions, etc. In this way, we could create a tool that would complement a wide variety of calculus textbooks, teaching styles and course structures.

The system should also contain case studies and other cross-curricular examples and links to help the student understand the interrelatedness of the material, and to remediate as necessary. These examples must be "live," that is, the student must be able to manipulate the mathematics in a meaningful way in order to get any benefit out of the exercises.

We wanted any modular system we produced to continue to use CASs for interactive mathematics. There are several important reasons for this. First, it's important to engage the student in the process of solving the problem. The computer should be used for number crunching, but should not be used as a substitute for reasoning through a problem. The student must be able to enter a wide variety of calculations in a freeform way, making errors and correcting them along the way, rather than being limited to changing only a small part of one equation. In a CAS, any type of calculation can be entered once the syntax of the system has been learned. Since CASs are designed to be flexible, they both require the student to enter equations themselves, and allow them to extend the problem, rather than limiting a student to the original example. Second, CASs are off-the-shelf solutions that are extremely good at what they do. This will save the multimedia developer from having to develop sophisticated mathematical or graphical tools themselves. Finally, there is a benefit to the student in learning how to use a professional tool at the same time as learning the concepts of calculus. When a student is introduced to a professional tool in this context, they will take away the skill of using it as well as the course content, both of which will carry them through their career.

Choosing the Authoring Tools and the Interface

The ideal situation would be to incorporate CAS files into a larger hypertext application, but typically this has been difficult to do in a way that is straightforward, portable cross-platform, and easily upgradable. Tools like Apple's Hypercard and Macromedia's Authorware, as well as some of presentation packages, such as Aldus Persuasion, offer ways to incorporate files by starting up a CAS on a hyperlink [7,8]. There are often problems in porting these applications across platforms, and they require a fair bit of sophistication on the part of the author regardless of the application. It is also very difficult to include new materials in these systems after publication, or to customize the structure of the material, since files are not maintained as separate entities in the package.

Mosaic publishing over a World Wide Web server offers an attractive solution. Mosaic provides a cross-platform interface which is both user-friendly and in common use, so the learning curve for both students and instructors should be short. The HyperText Markup Language (HTML) provides a simple way to format text and images nicely on screen, and to create hyperlinks to any type of file. Links can be made to navigate the course material, to connect related topics, and to seamlessly combine CAS files through the Viewer/Helper Application interface [9, 10]. Since HTML is easy to script, it is even possible that instructors could add their own problems, explanations and CAS files to the published material. The beauty of Internet publishing is that files retain their application identity, so each type of software is used to perform the function at which it is best.

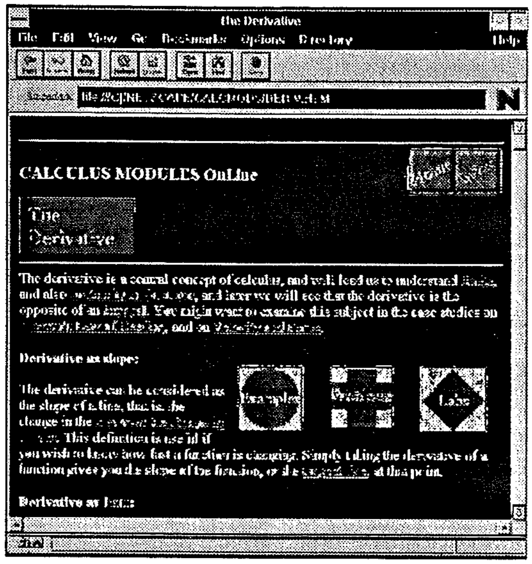


Figure 2

Figure 2 shows a screen from a preliminary version of *Calculus Modules On-line.* The prototype incorporates multimedia elements, hypertext, and CAS files under the Mosaic front-end, as described. There are hyperlinks (underlined text) to other sections of the "electronic book," which explain some of the terms used and guide the student to examples, problems and laboratories, and there are hyperlinked icons/buttons which lead to animated examples, and "live-math" CAS solutions, which are implemented in several computer algebra systems, allowing the instructor to choose which one the students should use.

Hyperlinks and Viewer Applications

This application depends not only on HTML's ability to link to other HTML documents, but also to external applications. One of the most powerful capabilities of Mosaic is it's ability to call up external pieces of software. Rather than build in the capability to run movies or play sounds, or any of the other host of multimedia headstands which might be required, the authors of Mosaic have simply allowed other applications to take over where Mosaic leaves off, which means the possibilities for combining external media are limitless. Mosaic can be configured to recognize any file extension as a particular Multipurpose Internet Mail Extension (MIME) type and subtype, and start the appropriate application on the user's machine when such a file is on the end of a link.

Some varieties of Web servers provide the MIME type of a file when it is served. This means that the MIME type for CAS files, and any others which are not HTML, should be specified both on the server (in the MIME configuration file) and within the Mosaic browser. Every time a link connects to a file with a .ms extension (the correct extension for Maple), Mosaic will launch the Maple application. Netscape Mosaic[11] currently provides the easiest way to set this functionality, using a dialog box from a pulldown menu. It is for this reason, as well as the extended set of HTML formatting available, that it is particularly recommended for viewing the *Calculus Modules.*

Customization and updating

Customization is a serious issue for many of PWS's customers, so it was important in the design of the *Modules* to incorporate plans for customization. This is reflected both in the "modularity" of the project -- the course sequence has been broken down into areas of the curriculum, and into case studies -- and in the individual file structure inherent in Mosaic publishing. If desired, we could, with very little effort, create a customized, hyperlinked "table of contents" which included only some modules, or some case studies. The system can therefore easily conform to any desired course organization.

In addition to customization, the prototype structure accommodates updating. As additional examples and modules are developed, they can easily be incorporated into the existing hyperlinked structure. It is our hope that the project will continue to grow and acquire good application examples and laboratories, contributed from the calculus teaching community at large. New modules can even be delivered by

downloading them from the Internet. In fact, entire system can be maintained and upgraded through this technique. It is our intention to maintain part of this project on our Web server, providing free access to the mathematics community, and encouraging them to participate in the evolution of the project.

It is important that the *Calculus Modules* are able to link to a number of different mathematical software packages from the same set of examples. As CASs gained popularity in the calculus community, most professors developed a preference for a particular system which they cling to with near-religious fervor. This is one way in which Calculus Modules are further customized. The instructor simply specifies to the student which computer algebra system to use (by specifying the icon, which is consistent throughout the Modules. Unwanted examples in other systems never appear unless the student selects those icons, and has the alternate CAS program available on their computer.

Tech Support and Other Open Issues

There are many problems remaining to be overcome in Mosaic and Web publishing. Since individual files are downloaded on demand from a server, large files will take a while to load to a student's machine, even from a school's local server. With text this is not a severe limitation, but with images, animations, and CAS files that contain extensive graphics, it can create a problem with a student's attention span. Our authors must be sensitive to this when creating anything for the Modules.

When considering viewers to include other file types, it's important to realize that the interface for external applications is not yet a fully developed one. Mosaic does not recognize when a viewer application is already open, since it does not have a fully API. The desired behavior upon encountering a specific file type would be for Mosaic to check for the application's state first, and only to open the specified document if the application were already open. A more sophisticated API would open the path to more sophisticated, and even more "interactive" collaboration between Mosaic's hypertext and the CAS.

An additional problem which is specific to mathematics and science instruction is the inability to format mathematics and tables in the currently popular set of HTML tags. The Arena[12] browser and HTML3 specifications promise some advances in that area, but at present there are few versions available, and no commercial versions.

Aside from the unresolved technical issues with this product, there are still a variety of opinions on how best to sell Web-related products. Our current plan is to site license the Modules and download the files to an individual school's server, where they would be kept behind a firewall. This does imply some installation and technical support work on the part of the publisher. In addition, this model assumes that the school has the appropriate viewer software on their network. Typically, this is the case, but no doubt there will be some exceptions.

Commercial Internet Publishing: the Future

Multimedia calculus modules can present mathematics in a way which is superior to a paper text, and can remove some of the tedium associated with mathematics classes. The system is based entirely on familiar, professional software, and will be relatively easy to use, adding clarity to the mathematics without adding complexity. The modules will allow the instructor or student to choose the order and content of the material, by following various hyperlinked trails, or by specifying customized "homepages", and by including additional links in the HTML pages. The instructor will even be able to choose from a variety of CAS files. All this can be accomplished at a cost to the publisher which does not greatly differ from that of publishing a book.

As more sophisticated hypertext-linked files and CAS applications are developed, calculus can be approached from the case study point of view, and modules can be developed across the science and engineering curriculum. This may result in an increased student interest in mathematics, a concept which is gaining popularity in the educational community[13].

In the future, developments in the API between Mosaic and viewers will be improved, and HTML display of mathematics will become available, making this system even more attractive as a development platform. Meanwhile, the existence of most of this technology in some form, and the ubiquitous nature of these tools, means that the *Calculus Modules* can be explored now in prototype form, and that contributions from the Calculus teaching community can be solicited, creating a richer, more solid product, which is the result of a

true community teaching effort.

The author would like to acknowledge the input, suggestions and continuing hard work of the current *Calculus Modules* development team: Dr. Frank Wattenburg at Weber State College, Dr. Don Hartig and Dr. Mike Colvin at California Polytechnic University, and Dr. Charles Patton at Hewlett Packard Corporation. Please contribute your own suggestions at our Web site:
`http://www.pws.com/pws/math/modules.html`.

Bibliography

[1] Sloan Conference at Tulane University on Calculus Reform, 1986, and Calculus Reform for a New Century Conference, Washington D.C., 1987.

[2] The Interactive Mathematics Text Project `gopher://imtp.math.upenn.edu`.

[3] *PWS Notebook Series for Mathematics*, PWS Publishing Company, Boston, 1994.

[4] Devitt, John S., *Calculus with Maple V*, Brooks/Cole, Pacific Grove, CA, 1993.

[5] Developed by Tim Berners-Lee while at CERN, Geneva, Switzerland. `http://info.cern.ch/hypertext/WWW/TheProject.html`.

[6]Mosaic software, National Center for Supercomputing Applications The University of Illinois, `http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/`.

[7]Russ, John, *Visualizations in Materials Science (CD)*, North Carolina State University, Raleigh, NC, 1994. `http://vims.ncsu.edu/`

[8] Jonassen, David and Wang, Sherwood, "The Physics Tutor: Integrating Hypertext and Expert Systems," *J. of Ed. Tech. Syst.*, Vol. 22(1), pp. 19-28, 1993-94.

[9] Phil Smith, Robert Curtis, and Chris Barker, *calculus@internet*, San Joaquin Delta College, CA, `http://mac205.sjdccd.cc.ca.us/cai-home.html`.

[10] Andy Elby and Paul Manly, *The Interactive Physics Problem Set*, UC Berkeley, CA, `http://info.itp.berkeley.edu/Vol1/Contents.html`.

[11] *Netscape* software, Netscape Communications Corporation, `http://home.mcom.com/`.

[12] The Arena browser from the World Wide Web Consortium, `http://www.w3.org/hypertext/WWW/Arena/`.

[13]Shank, Roger and Cleary, Chip, *Engines for Education*, Institute for the Learning Sciences, Northwestern University, Evanston, IL, 1994. `http://www.ils.nwu.edu/~e_for_e/`.

About the Author

Leslie Bondaryk has a B.S. and M.S. in Electrical Engineering, from MIT and UC Santa Barbara, respectively. She has been actively involved in the creation and development of educational materials in technical subjects since 1986, working with authors at universities, in industry and generating her own material. She worked at MathSoft, Inc., for two years, creating electronic books in math, science and engineering subjects in Mathcad. While there, she piloted the *Schaum's Interactive Outline Series* with McGraw-Hill, and was series editor for that project while at MathSoft. She is currently the Manager of Technology Development at PWS Publishing Company, where she continues to ply her skills in programming, computer algebra systems, and multimedia development. She is participating in the innovation of the next generation of technology educational tools, including the online calculus project described here.

Electronic Publishing of Virus Structures in Novel, Multimedia Formats on the World Wide Web

Stephan M. Spencer and
Jean-Yves Sgro
Institute for Molecular Virology,
University of Wisconsin-Madison,
Madison, WI 53706

Abstract

Visualizations of complex biological structures such as viruses are well-suited to distribution via the electronic medium of the World Wide Web, complementing the peer-reviewed publication of figures in scientific papers. Animation and color can be employed to accentuate particular features of structure, and thus a greater information content can be imparted than would be possible with printed media. Structural information that is easily accessible in a standard, meaningful, and even interactive format can be an effective tool in teaching and research. We at the Institute for Molecular Virology, University of Wisconsin-Madison have developed a World Wide Web server † *which disseminates structural information in novel formats world-wide to scientists, teachers, students, and the public.*

Animations, interactive models, and high resolution color-coded images of viral particles and proteins are available, many of them exclusively, from our site. To create comparable visualizations using generally available resources would prove difficult, to a large extent because of the complexity of these structures which would require specialized computing equipment, a great deal of computing expertise, and datasets that are either not publicly available or that need to be reconstructed by symmetry operations from the PDB coordinates (which represent one sixtieth of the complete particle). The previously rendered images and animations, however, can be readily downloaded and viewed on personal computers connected to the Web.

We have used animation and false color extensively to present structural details that may not have been visible without additional figures, such as orthogonal or radial sections, multiple views, or stereoscopic images. Coloring the virus particle according to the protein subunits (Fig. 1a) allows the viewer to determine the composition of notable structural elements on the particle surface. Radial depth cueing [1] (Fig. 1b) is a technique for applying false color that correlates with the radial distance from the center of the particle. We often use these coloring techniques in conjunction with animation techniques. Both spin animation, i.e. rotation of the particle around an axis (Fig. 2a), and radial depth cueing are effective in enhancing the surface topography and improving the presentation of peaks, canyons, and pores. The cropping of frontal (Fig. 2b) or radial (Fig. 2c) sections reveals internal features.

We offer yet another useful representation of virus crystal structure data, one that takes advantage of the capability of the WWW protocol to "view" atomic coordinate files interactively [2]: a three-dimensional model of an icosahedral asymmetric unit of the virus, displayed in the context of the icosahedral framework (Fig. 3). These interactive models employ the KineMAGE [3] molecular graphics program as a helper application that needs to be installed on the user's computer. We also anticipate offering "navigable" QuickTime [4] movies of rendered virus structures in the near future, which will provide even greater flexibility by alleviating the requirement of a molecular graphics helper program, yet still allowing the real-time manipulation of these structures in three dimensions.

This type of electronic publishing has marked advantages over a CD-ROM because it can be updated instantly. Unlike a CD-ROM, performance is affected by Internet bandwith limitations, namely the type of connection and overall Internet traffic. Once the animation or structural file has been transferred, however, all manipulation becomes local to the user's machine, and thus these visualizations truly offer real-time interactivity.

These virus visualizations enhance conventional virology instruction by offering unique resources to students and teachers. Animated or interactive visualizations of viruses allow students to interact in new ways with the course material and can supplement

---

† `http://www.bocklabs.wisc.edu/Welcome.html`

traditional teaching aids such as textbooks and lectures. With advances like the World Wide Web protocol and Kinemage, electronic publishing of virus structures have become decreasingly less platform-dependent and thus the are now accessible to a much wider audience. In addition to these visualizations, we provide other course materials on our server, such as virology tutorials, course notes, syllabi, and journal articles. This material is most effectively assembled into a coherent whole by the teachers who are on the 'front lines,' not by us as electronic publishers. To achieve this end, we have designed a fill-in form interface that allows instructors without any knowledge of HTML (HyperText Markup Language) [11] to create clickable course outlines ("hypersyllabi") which are maintained on our server. This coupled approach of providing useful information in unique, multimedia formats and a dynamic environment for organizing the information will, we believe, enhance distance education and collaborative teaching.

Acknowledgments

We gratefully acknowledge Jack E. Johnson and colleagues at the Structural Biology Group at Purdue University for supplying us with the atomic coordinates that were used to generate Figure 1b and Timothy S. Baker and colleagues also at Purdue University for supplying us with the three-dimensional cryo-electron microscopy dataset that was used to generate Figure 2. This work was performed at the Institute for Molecular Virology, University of Wisconsin-Madison, with partial funding provided by the Lucille P. Markey Charitable Trust. Work in establishing the WWW Server for Virology was used to fulfill part of the requirements for awarding the Masters of Science degree in Biochemistry (5-95) to S.M.S. for work he performed in the laboratory of Prof. Max L. Nibert.

References

1 Grant, R.A., S. Cranic, and J.M. Hogle. 1992. Radial depth provides the cue. *Current Biol.* **2**:86-87.

2 Rzepa, H.S., B.J. Whitaker and M.J. Winter. 1994. Chemical Application of the World-Wide-Web *J. Chem. Soc. Chem. Commun.* 1907

3 Richardson, D.C. and J.S. Richardson. 1992. The kinemage: a tool for scientific communication. *Prot. Sci.* **1**:3.

4 Apple Computer, Inc. Cupertino, CA.

5 Berners-Lee, T., and D. Connoly. 1993. Document Type Definition for the HyperText Markup Language as used by the World Wide Web application (HTML DTD). *IETF Internet Draft.*

6 Rossman, M.G., E. Arnold, J.W. Erickson, E.A. Frankenberger, J.P. Griffith, H.-J. Hecht, J.E. Johnson, G. Kamer, M. Luo, A.G. Mosser, R.R. Rueckert, B. Sherry, and G. Vriend. 1985. Structure of a human common cold virus and functional relationship to other picornaviruses. *Nature* **317**:145-153. (PDB entry # 4RHV)

7 Connolly, M.L. 1993. The molecular surface package. *J. Mol. Graphics* **11**:139-141

8 Fisher, A.J., B.R. McKinney, J.-P. Wery, and J.E. Johnson. 1992. Crystallization and preliminary data analysis of Flock House virus. *Acta Crystallogr.* Sect. B **48**:515-520.

9 Colloc'h, N. and J.-P. Mornon. 1990. A new tool for the qualitative and quantitative analysis of protein surfaces using B-spline and density of surface neighborhood. *J. Mol. Graphics* **8**:133-140.

10 Ferrin, T.E., C.C. Huang, L.E. Jarvis, and R. Langridge. 1988. The MIDAS display system. *J. Mol. Graphics* **6**:13-27.

11 Silicon Graphics, Inc. Mountain View, CA.

12 Dryden, K.A., G. Wang, M. Yeager, M.L. Nibert, K.M. Coombs, D.B. Furlong, B.N. Fields, and T.S. Baker. 1993. Early steps in reovirus infection are associated with dramatic changes in supramolecular structure and protein conformation: analysis of virions and subviral particles by cryoelectron microscopy and image reconstruction. *J. Cell Biol.* **122**:1023-1041.

13 Iris Explorer. Silicon Graphics, Inc., Mountain View, CA

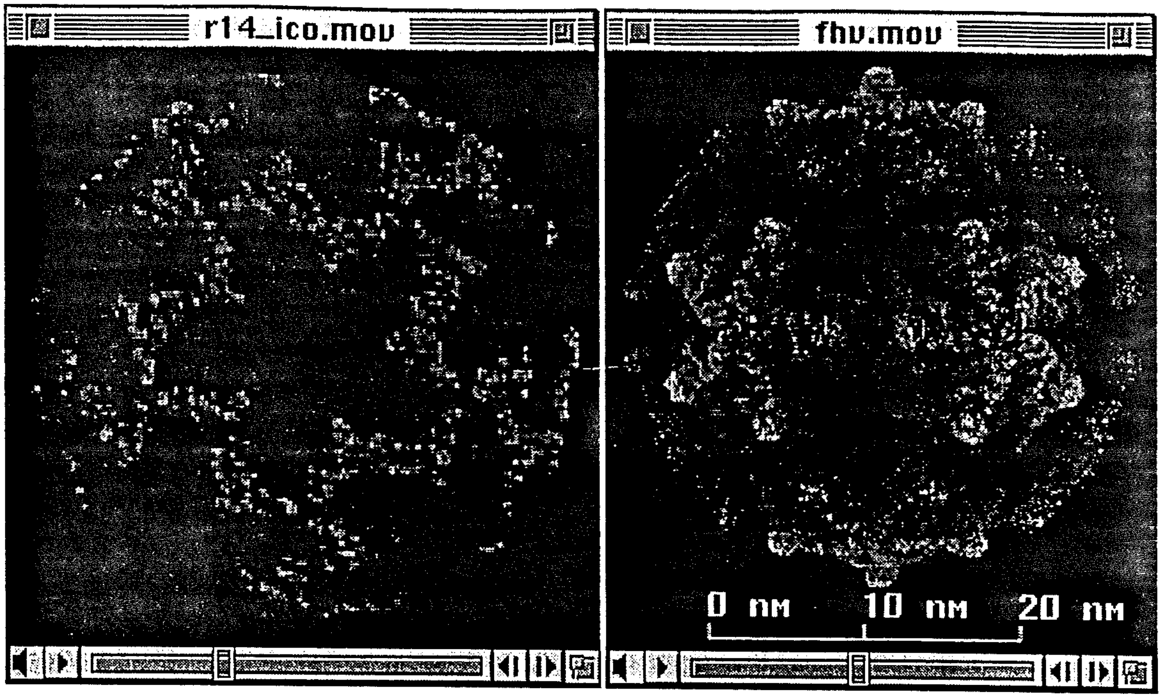


FIGURE 1: Two types of false color are applied to virus structures. a.) Color corresponds to the protein subunit. Human rhinovirus 14 (a common cold virus) [6] ; VP1 (Viral Protein 1) is colored blue, VP2 green, and VP3 red (VP4 is inside and not visible). Rendered using srf [7] on a Silicon Graphics workstation. b.) Color is a function of the distance from the center of the particle, i.e. radial depth cueing [1]. Flock house virus (an insect virus) [8]. Rendered using Spline [9] and MIDAS–Plus [10] on a Silicon Graphics [11] workstation.

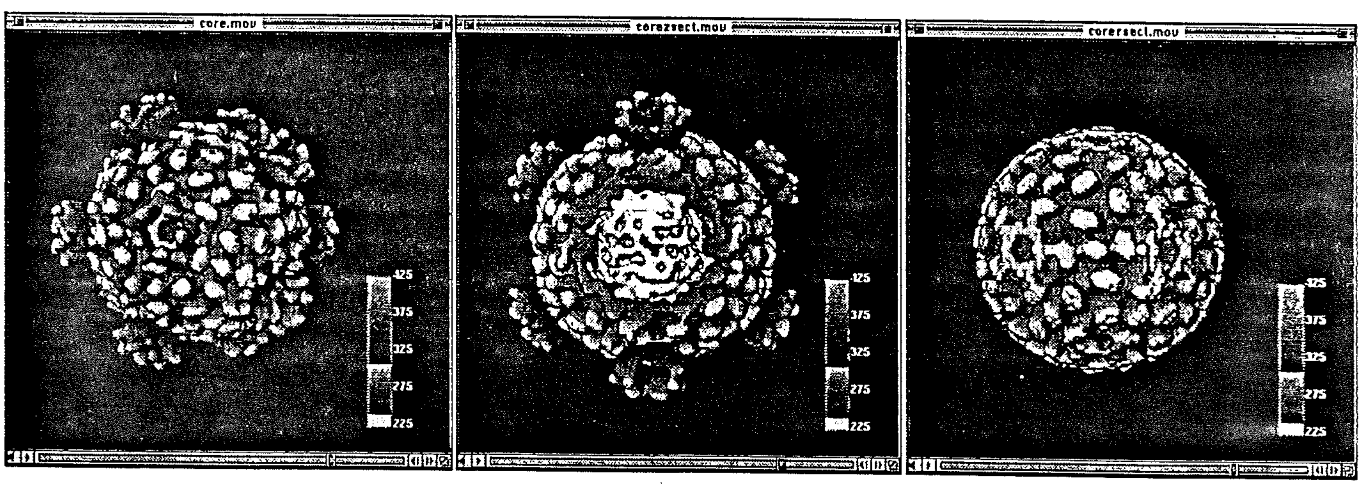


FIGURE 2: Three types of animation were used to display the virus structures. The core particle of mammalian reovirus [12], shown with radial depth cueing. a.) Rotation around an axis (spin animation). b.) Cropping in the z–direction. c.) Radial cropping. Rendered using Iris Explorer [13] on a Silicon Graphics workstation.

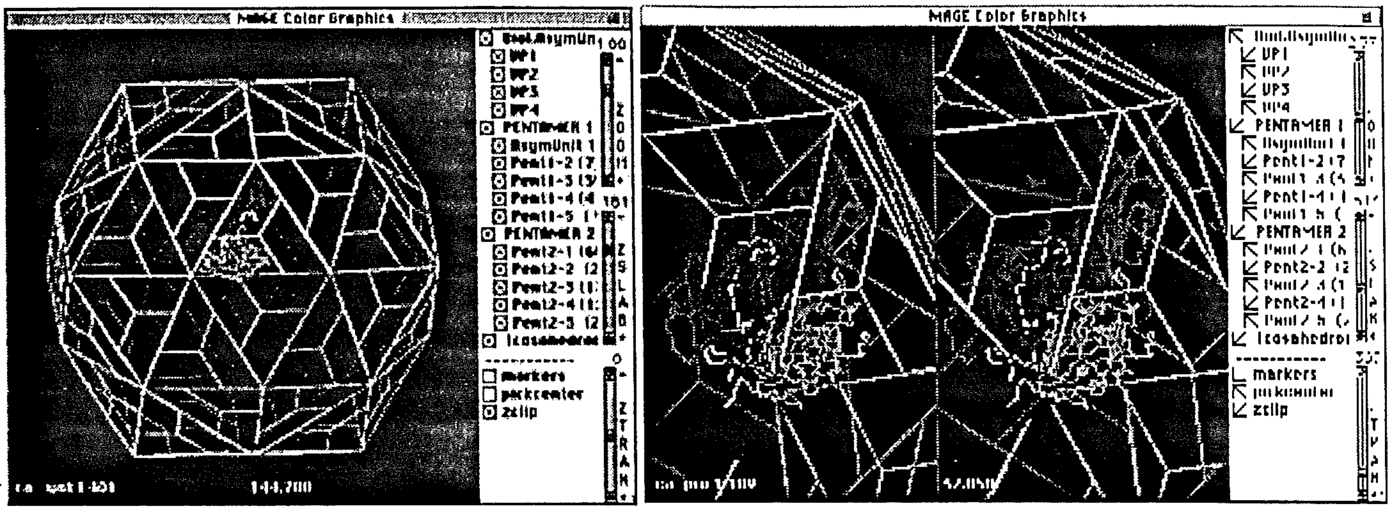


FIGURE 3: The icosahedral asymmetric unit of human rhinovirus 14, viewed interactively on a Macintosh [4] using the molecular graphics program KineMAGE [3]. a.) Normal view. b.) Stereo view.

Language Learning via World Wide Web

Mark H. Nodine*

Motorola Cambridge Research Center
One Kendall Square, Building 200
Cambridge, MA 02139

Abstract

Learning foreign languages is a task that is difficult and time-consuming. This paper describes a course on the World Wide Web for teaching the foreign language Welsh, and the tools which enabled the development of that course. The Uniform Resource Locator for the course is `http://www.cs.brown.edu/fun/welsh`.

1 Introduction

Learning foreign languages is a task that is difficult and time-consuming. While it is possible to learn a foreign language by simply picking up a book on the subject, there are many difficulties that confront the learner who would attempt to do so. The course in Welsh hosted by the Brown University Computer Science Department is an experiment to see how effectively a foreign language can be taught using multimedia hypertextual tools via the Internet.

In Section 2, I give the relevant background information concerning the World Wide Web. Section 3 gives some of the features of the Welsh course. Section 4 describes the relevant technology that enabled the course and lists the specific tools that were developed. The conclusions and usage statistics appear in Section 5.

2 The World Wide Web

The Internet gives the potential for very widespread dissemination of information. A particular boon to information-finding via the Internet was the development of the World Wide Web (WWW). The WWW was started by CERN to provide a distributed hypertext environment.[2] In a hypertext document, certain words are highlighted as links to other parts of the same document or to other documents; clicking on those words traverses the link. In this way, it is straightforward to browse large quantities of information. The WWW specifies the destination of a hypertext link using a Uniform Resource Locator (URL).[8] A URL contains two primary parts: the first part (up to the first colon in the string) specifies the protocol to be used to retrieve the information; the second part (everything else) gives the specific information needed by the protocol. Some common protocols used in URLs are the HyperText Transfer Protocol (HTTP), the File Transfer Protocol (FTP), and the Gopher protocol.

The World Wide Web is built around a distributed client-server model. In this model, the Web user ("surfer" in Internet jargon) runs a local client that connects to a server using one of the WWW protocols. There are many different client programs (e.g., xmosaic, NCSA Mosaic, emacs-19) and even several different servers. The user interface is provided by the client. The supported protocols also differ from client to client, although the core set of protocols is extensive.

One of the features of HTTP is that files of different types can be transferred using it. This polymorphism is accomplished by using the Multi-purpose Internet Mail Extensions (MIME) protocol[1] to specify the type of the file being transferred. The MIME protocol was developed to allow different kinds of files to be transferred through the ASCII-only medium of electronic mail, and appends to

*E-mail: nodine@mcrc.mot.com.

the mail header information indicating (among other things) that it is a MIME document, what kind of document it is, and what kind of encoding it uses. The document types have at least two parts, the first of which indicates the general type of file, and subsequent parts getting more specific. For example, "text/plain" stands for plain text documents, "application/postscript" for Postscript documents, and "image/gif" for GIF image files. When a MIME-compliant mail reader receives a message that is some type other than "text/plain", it can check whether an external viewer is available for that kind of document, and if one exists, it can spawn it off, allowing the end user access to the file. Because HTTP builds upon the MIME interface, which supports multimedia through the external viewer capability, it also supports multimedia.

The most common type of file accessed through HTTP is a HyperText Mark-up Language (HTML) file. HTML is a textual layout language that contains markup elements for headings, text styles, links, forms, and other things.[3] The HTML standard is evolving; most clients currently support HTML level 2, while development work continues on HTML level 3 (also known as HTML+).[7]

A second feature of HTTP that greatly expands its flexibility is the ability to interface with computation engines using the Common Gateway Interface (CGI).[6] These computation engines are frequently used to provide the capability of searching databases.

3 Features of the Welsh Web Course

What I have written can be considered a textbook for an introductory course in the Welsh language. My own experience with the Welsh language has been from teaching it to myself using Jones,[5] so I am well aware of the pitfalls that confront someone seeking to do likewise. Accordingly, the book contains the following features:

1. A *table of contents* for quick referencing. Each individual lesson also has a local table of contents.

2. A *glossary of grammatical terms* for those with no prior language-learning experience. This glossary describes terms such as "noun" and "relative clause".

3. *Welsh-English* and *English-Welsh* lexicons. Initially these were intended to contain only the words used in the lessons, but they have since been substantially augmented.

4. Both lexicons contain *notes* describing peculiar considerations for words, such as irregular pronunciation or which prepositions verbs govern. The English-Welsh lexicon contains additional notes to distinguish between various possible Welsh translations of words.

5. Both lexicons contain many *phrases.*

6. A very complete *index.* One of the difficulties with Jones[5] is that it is quite difficult to locate things in it. I made certain that the index was complete to avoid this problem.

7. Considerable *humor* to enliven the discussions.

8. *Footnotes* "for the terminally curious", to accommodate the learning styles of both those who want to know all the details at once and those who are intimidated by extraneous information.

9. *Conversations* and *exercises* to illustrate and practice using the grammatical constructs, along with translations and answers.

10. *Appendices* that provide reference materials such as conjugations of irregular verbs.

11. All the lessons are available for anonymous FTP in a *formatted ASCII form.*

So far, all of the items listed are such as could be found in any good textbook that is printed on paper. However, the materials also take advantage of the capabilities of HTML and MIME:

1. There are numerous *hypertext links* to cross-references, footnotes, and bibliographic references. There are perhaps 50 of these links per chapter. The table of contents and index also contain links to the appropriate sections.

2. The chapter on pronunciation has links to *sound files*, so that learners with the appropriate equipment can actually hear how the words are supposed to sound.

3. Since the document is dynamic, links to a *revision history* are available for each file. There is also a revision history by date, so that users can find out all the changes that have occurred since they last visited the course.

4. Since the document is public, an *access log* is available giving such statistics as how many accesses the course has received, which countries are most active in accessing it, and which files receive the most accesses.

5. There is a *searching lexicon* as shown in Figure 1 implemented with CGI. Having a searching lexicon is more than a minor convenience in Welsh; it can frequently make the difference between finding a word and not because of the following characteristics of Welsh:

   (a) Welsh words are subject to mutations that change the initial consonant(s) of the word. There are three different schemas for mutation, and the back-conversion is often not unique. For example, the word "fyny" could be listed under "f", "b", "m" or "g".

   (b) Changing the stress of a Welsh word, for example in the formation of plural nouns, often changes the internal vowels, so that "ceir" (cars) needs to be looked up under "car" (car).

   (c) The Welsh alphabet uses two letters (digraphs) to represent some sounds; these digraphs are alphabetized as separate letters and have their own order in the alphabet. For example, the word "anghofio" (forget) is alphabetized between "agos" (near) and "ai" (either) since the digraph "ng" comes between "g" and "h" (which precedes "i" as in English). To make matters worse, "ng" is sometimes a digraph (filed after "g") and sometimes not (filed with the "n"'s).

   (d) Welsh verbs and many adjectives and prepositions have inflected endings.

   (e) Welsh speakers often drop parts of words, leaving behind only an apostrophe (if that) to indicate something is missing. A common pattern of droppage is to omit the first syllable.

The searching lexicon, when used in Welsh-English mode, attempts to undo any mutations, and also recognizes conjugated forms of verbs (even irregular ones), adjectives, and prepositions, as well as plurals of nouns. Wildcards can be used in "Partial Word" mode so that you can replace any apostrophes by a "*" and see what you find.

4 Technology of the Welsh Web Course

The Welsh course began after the creation of the WELSH-L LISTSERV list at IRLEARN.UCD.IE established a forum for people to exchange messages in and about the Welsh language. There were many new subscribers to the list who did not know Welsh, but who had an interest in learning it. I was eventually pursuaded to put together a beginning Welsh course to distribute through the mailing list. Since there was no way to be certain that the readers would have MIME-compliant mail readers, nor that any particular external viewer would be universally available, I decided to distribute the course in an ASCII-only format. At about this time, I became aware of Feldman's work on a "structure-enhanced text" format known as setext.[4] A structure-enhanced text document contains "typotags" for mark-up, but attempts to do so in a way that is unobtrusive to the text, and using elements that people automatically interpret correctly. Table 1 summarizes the relevant setext typotags. Text that is indented any amount other than two spaces is considered verbatim text ("preformatted" in HTML parlance). Figure 2 gives an example setext. The advantage of using a format like setext was that it allowed the course to be distributed through an ASCII medium, while still retain-

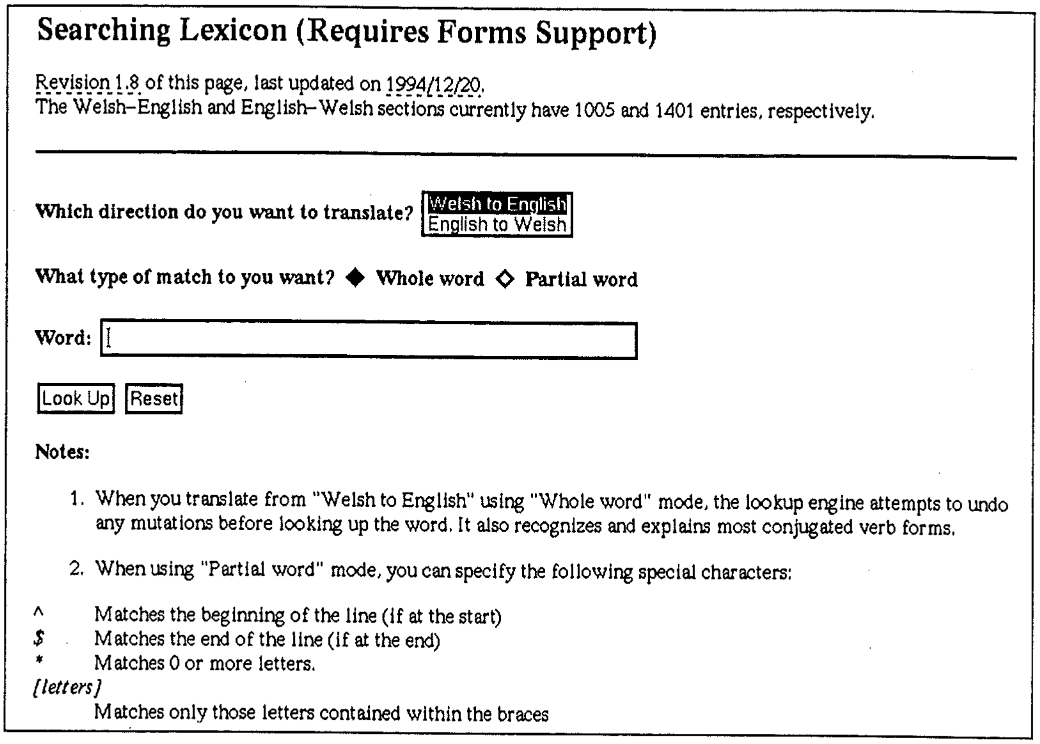


Figure 1: NCSA Mosaic for X rendering of the searching lexicon.

ing the formatting information needed to convert to HTML. Posting the course in ASCII to the WELSH-L mailing list allowed me to receive comments and corrections from many people whose Welsh is better than mine, many of them native speakers, who acted as a safety net for me.

The remainder of this section describes the tools I developed to support the course. All of the tools are in Perl for maximum portability.

4.1 etx2html

One of the tools I developed[1] converts setext files (which are files with extension .etx, for Enhanced TeXt) to HTML; Figure 3 shows the results of converting the example setext from Figure 2 into HTML. This 400-line Perl script automatically creates a clickable table of contents at the top of the file and puts a signature at the bottom (there are command-line options to turn these off). This tool is useful enough that we have found many applications for it within the Motorola Cambridge Research Center other than the Welsh course.

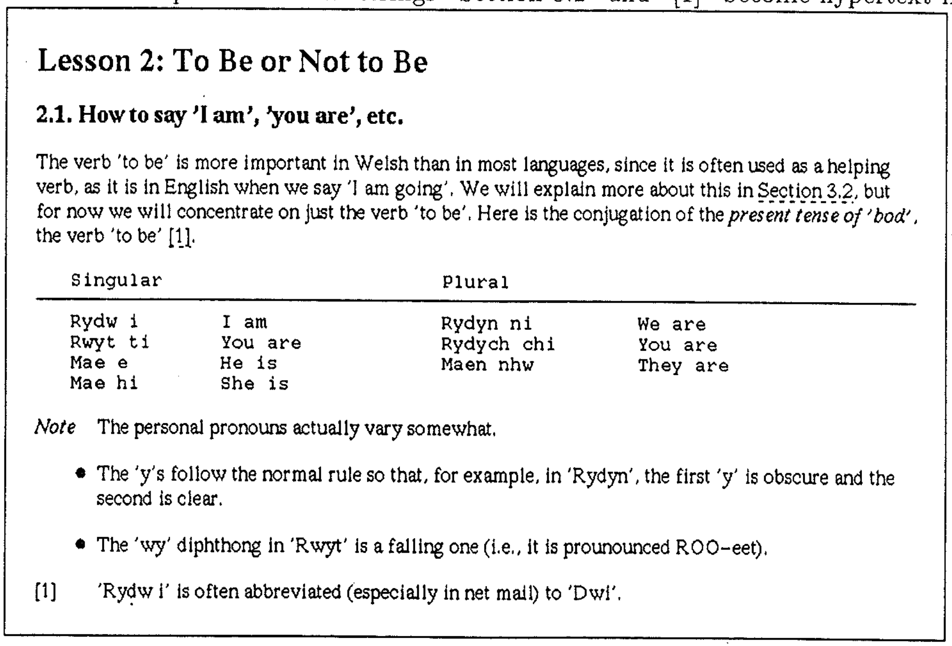
4.2 dtx2etx

It soon became apparent that while most of the text was to be identical between the HTML and the setext versions, there were certain items to be included in the HTML version only (such as a row of buttons at the beginning and end of the page) and certain items to be included in the etx version only (such as the pointer to where to access the HTML files). Additionally, getting the format of the etx files exactly correct was subject to error, since the last character of the title or subtitle underline had to line up exactly with the last visible character of the title or subtitle itself, and it was necessary to remember to indent all body text by exactly two spaces. Accordingly, I developed a file format called "dtx", which doesn't really stand for anything but seemed to be a reasonable predecessor for "etx". The 160-line `dtx2etx` converter is very simple. Anything that looks like a bulleted list, a footnote, a comment (including hypertext definitions) or a numbered list is passed through unchanged. A line of the

[1] This tool actually began its life at the hands of Tony Sanders, but was almost completely rewritten by me and substantially embellished.

```
Lesson 2: To Be or Not to Be
============================

2.1. How to say 'I am', 'you are', etc.
---------------------------------------

  The verb 'to be' is more important in Welsh than in most languages,
  since it is often used as a helping verb, as it is in English when we
  say 'I am going'.  We will explain more about this in Section_3.2_,
  but for now we will concentrate on just the verb 'to be'.  Here is the
  conjugation of the _present_tense_of_'bod'_, the verb 'to be' [1]_.

   Singular                        Plural

   ---------------------------------------------------------
   Rydw i       I am               Rydyn ni         We are
   Rwyt ti      You are            Rydych chi       You are
   Mae e        He is              Maen nhw         They are
   Mae hi       She is

Note: The personal pronouns actually vary somewhat.

* The 'y's follow the normal rule so that, for example, in
  'Rydyn', the first 'y' is obscure and the second is clear.

* The 'wy' diphthong in 'Rwyt' is a falling one (i.e., it is
  prounounced ROO-eet).

[1] 'Rydw i' is often abbreviated (especially in net mail) to 'Dwi'.
.. _Section_3.2 Lesson03.html#3.2. How to say 'I am reading'
```

Figure 2: An example setext. The strings "Section 3.2" and "[1]" become hypertext links.



Lesson 2: To Be or Not to Be

2.1. How to say 'I am', 'you are', etc.

The verb 'to be' is more important in Welsh than in most languages, since it is often used as a helping verb, as it is in English when we say 'I am going'. We will explain more about this in Section 3.2, but for now we will concentrate on just the verb 'to be'. Here is the conjugation of the *present tense of 'bod'*, the verb 'to be' [1].

Singular		Plural	
Rydw i	I am	Rydyn ni	We are
Rwyt ti	You are	Rydych chi	You are
Mae e	He is	Maen nhw	They are
Mae hi	She is		

Note The personal pronouns actually vary somewhat.

- The 'y's follow the normal rule so that, for example, in 'Rydyn', the first 'y' is obscure and the second is clear.
- The 'wy' diphthong in 'Rwyt' is a falling one (i.e., it is prounounced ROO-eet).

[1] 'Rydw i' is often abbreviated (especially in net mail) to 'Dwi'.

Figure 3: NCSA Mosaic for X rendering of the HTML generated from the setext of Figure 2.

Function	Setext Representation	Effect
Title (≤ 1 per text)	`^Title` `^=====`	A title in the chosen style†
Heading (≥ 1 per text)	`^Subhead` `^-------`	A subhead in the chosen style†
Body text	66-char lines indented 2 spaces	Lines undented and wrapped
≥ 1 bold words	`**bold words**`	Emboldened words
≥ 1 italic words	`~italic words~`	Italicized words†
≥ 1 underlined words	`_underlined_words_`	Underlined words
≥ 1 hypertext words	`hypertext_words_`	Hypertextual link
Bulleted text	`^*␣Text`	• Text
Hypertext definition	`^..␣_hypertext_words␣URL`	Defines address
Comment	`^..␣Any comment here`	Line hidden
Logical end of text	`^..$`	Taken note of
Horizontal rule	25+ underlines	Horizontal rule
Definition	`^Data tag:  Data definition`	Definition list
Footnote	`^[number]␣Text`	Definition list†
Numbered list	`^(number)␣Text`	Definition list

Table 1: Typotags for structure-enhanced text. Here the caret character stands for the beginning of the line, the dollar sign for the end of the line, and the square cup for a space character; they are not typed literally. The last four functions are my enhancements relative to the standard setext definition. Some of the irrelevant typotags in the standard setext definition have been left out of this table. The typotags labeled with † create targets for hypertext links.

form "`=␣Title`" is converted to a setext title and a line of the form "`-␣Subtitle`" is converted to a setext subtitle. Anything else has 2 spaces prepended to it, so that body text begins at the left margin and anything indented 1 or more space becomes preformatted text.

The `dtx2etx` converter also has command-line options for omitting all hypertext references and definitions (the links get turned into normal text) and for selectively ignoring parts of the file. There are three parts of the file that can be ignored: the part preceding the title, the part following the end of text line (".."), and the part between the title and the first subtitle. The first two of these sections are omitted from the etx conversion; the last from the HTML. This converter is also in heavy use at the Motorola Cambridge Research Center.

4.3 `counters`

It is useful to be able to define cross-references symbolically, so that, for example, if a new footnote is inserted, all the footnotes get renumbered automatically and any references remain correct. To allow this cross-referencing, I implemented a 260-line Perl script. There are two kinds of directives that `counters` processes: those contained within matching $'s and those within matching #'s. The former are definitions of variables and the latter are (mostly) references to variables (there is one hybrid operation in the latter category that is a combination definition and reference).

Definitions assign a string value to a variable. The syntax for definitions is:

```
$var: string $
$var=string$
```

The first form (involving ":") is compatible with RCS automatic variables; any leading or trailing spaces around the string are ignored. In the second form, all characters are significant. The second form of definition may also have references (described below) in the string. Variables have the format

```
var := [counter.]tag
```

where the `counter.` part is optional.

References get substituted with the (possibly modified) value of the variable. The syntax for references is:

```
#var#
#var/pat/subst/mods#
#+counter.tag#
```

The first form is a simple substition of the variable string. The second form does a pattern match on the variable string and returns the substituted form (or the original form if the pattern did not match). `pat`, `subst`, and `mods` are in Perl-style. The third form is a hybrid form. It increments the counter and assigns the resulting value to the variable `counter.tag`. Incrementing the counter means adding 1 to the last number present in the counter, or appending 1 if there was no number present.

The fact that the counters are compatible with RCS automatic variables and that substitutions could be applied made it possible to define references that look the same in any file but go to the correct "Next" or "Previous" file automatically by using the file name as the basis for computing the destination.

4.4 xref

The index is collected using a 175-line Perl script called **xref**. This script runs through all the files named on its command line looking for setext italic typotags (within tildes), even those in comments. Any such tags that it finds become index entries. The **xref** utility can also be run on a single file to produce a list of symbols exported by that file for inclusion in another file.

4.5 changelog

The 160-line **changelog** script runs over the source files and pulls out their revision history from the RCS change logs. It can create revision histories that are indexed either by date or by file. This script has been used here at the Motorola Cambridge Research Center to create a revision history for the source of an experimental parallelizing C compiler we are building.

4.6 wdlookup

The **wdlookup** script is a 630-line Perl script to search the Welsh dictionaries. It is the command executed at the heart of the searching lexicon, the one that understands the various verb forms and irregular verbs. Here is the output of wdlookup when used to look up the word "haf":

```
haf [-au, m.] - (n.) summer
hau [irreg.] - (v.) sow, disseminate
  { 'haf' is the word 'hau' in the 1.s. pres. ind. }
```

Notice that it provided an explanation for why it matched "hau". This command is actually more flexible than would be needed "simply" for searching the lexicons, since it has one more mode of operation: as a Welsh spell-checker. When invoked with the right set of command-line options, it scans one or more files and prints out any words it cannot find in its dictionary. This process is useful in two contexts: (1) I can pass the conversations through it to make sure that I have actually defined all the vocabulary necessary to understand them, and (2) as I receive Welsh messages from WELSH-L, I can pass them through **wdlookup** to create a list of words for augmenting the lexicons.

4.7 transpose

The 135-line **transpose** script takes a list of vocabulary words going from Welsh to English and converts them to a list going from English to Welsh. It is careful to keep information dealing with parts of speech and inflected forms.

4.8 alphabetize

To enable automatic incorporation of new words into the lexicon, I wrote the 310-line script **alphabetize**. Recall from the earlier discussion that alphabetizing Welsh is not a trivial process because of the digraphs. The alphabetizer also provides various warnings, such as phrases that are defined inconsistently or multiple definitions for words. When alphabetizing on the English side, expressions are filed under each of the constituent words, so that

```
for the sake of - (prep.) er mwyn
```

is filed under "for", "sake", and "of" (it has a kill list of unimportant words which eliminates "the").

5 Conclusions

The URL `http://www.cs.brown.edu/fun/welsh` provides access to the first seven lessons, three appendices, the glossary of grammatical terms, the lexicons, the index, the change logs, the access log and references. The course has received almost 32000 accesses since June, with each month's count surpassing that of the previous month. 18 distinct hosts have accessed the course more than 100 times, a sign of serious interest in learning the language. Of the hosts whose nationality can be traced, 39 different countries are represented: 60% in the US and 20% in the United Kingdom. While it is still too early to tell how well people are learning Welsh from the course, the enthusiastic response makes it clear that many people are learning some Welsh from it.

It might be objected that many of the tools developed were unnecessary, since the HTML could have been developed either with a WYSIWYG[2] HTML editor, or using a tool such as LaTeX which already has built-in support for cross-referencing and can be converted to HTML. However, there are at least three reasons why this approach would not have been as effective:

1. The resulting ASCII version of the lessons, if one could be obtained at all, would not be formatted very nicely and would likely have a lot of extraneous text.

2. It is unlikely that any WYSIWYG tool provides the confluence of capabilities needed for the Welsh course. Operating on plain text files made it easy to combine tools so that, for example, the alphabetizer, the lookup engine, and the HTML generation all understand the same format for the lexicons.

3. It is helpful to be able to edit the course from a plain terminal, which would not be possible using FrameMaker or another WYSIWYG editor. The fact that all the source files are plain text is what permits things like the automatic generation of change logs.

[2]What You See Is What You Get.

6 Acknowledgements

This course could not have been done without help. I want to thank Roger Vanderveen for getting me started with the development of the course. Briony Williams recorded the sounds and made many helpful comments, especially with respect to the pronunciation of the language. Geraint Jones, a native Welsh speaker, made many corrections to ensure that I was not leading the students awry. Finally, many members of the WELSH-L mailing list provided input.

References

[1] N. Borenstein and N. Freed. MIME (Multipurpose Internet Mail Extensions) part one: Mechanisms for specifying and describing the format of Internet message bodies. Technical Report RFC 1521, Bellcore, Innosoft, September 1993. `http://www.cis.ohio-state.edu/htbin/rfc/rfc1521.html`.

[2] Thomas Boutell. World Wide Web FAQ. World Wide Web, 1994. `http://sunsite.unc.edu/boutell/faq/www_faq.html`.

[3] Daniel W. Connolly. HTML specification review materials. World Wide Web, December 1994. `http://www.hal.com/users/connolly/html-spec/HTML_TOC.html`

[4] Ian Feldman. Setext information and samples. World Wide Web, 1994. `http://www.bsdi.com/setext`.

[5] T.J. Rhys Jones. *Teach Yourself Living Welsh.* Hodder and Stoughton, Kent, England, 1977.

[6] Rob McCool. The common gateway interface. World Wide Web. `http://hoohoo.ncsa.uiuc.edu/cgi/`.

[7] Dave Raggett. HTML+ (hypertext markup format). World Wide Web, November 1993. `http://www11.w3.org/hypertext/WWW/MarkUp/HTMLPlus/htmlplu`

[8] Uniform resource locators. World Wide Web. `http://www11.w3.org/hypertext/WWW/Addressing/URL/`.

The Design of MMM: A Model ManageMent System for Time Series Analysis

Oliver Günther and Rudolf Müller
Institut für Wirtschaftsinformatik
Humboldt-Universität zu Berlin
Spandauer Str. 1, D-10178 Berlin, Germany
{guenther,rmueller}@wiwi.hu-berlin.de

Andreas S. Weigend
Department of Computer Science and
Institute of Cognitive Science
University of Colorado
Boulder, CO 80309-0430, USA
andreas@cs.colorado.edu

Abstract

Time series analysis and prediction is turning into an interdisciplinary subject where data and methods are being contributed from a broad variety of disciplines, including economics, physics, computer science, and statistics. Model management systems were originally designed for operations research applications. With thousands of methods and gigabytes of data now available on the Internet, however, such systems may become a crucial component for the efficient organization and exchange of any computer-based work in these areas. This paper introduces the model management system *MMM* that combines model management with the *World Wide Web (WWW)* to provide an infrastructure for interdisciplinary, worldwide distributed research on time series analysis. In particular, MMM will provide a platform to make related research results applicable and verifiable.

1 Introduction

With the increasing availability of high-capacity wide area computer networks, the sharing of data among distributed teams is becoming a matter of course in numerous professions. Scientists use the Internet to exchange experimental data and to write joint papers with co-authors at remote locations. Industry uses company networks (or the Internet as well) to facilitate cooperative work between teams at different sites. More and more institutions use the *World Wide Web (WWW)*, an Internet-based information system, to post information about themselves and, in turn, to gather data on just about any topic of interest from data sources around the world.

While the Web and its clients (such as *Mosaic*) made it a lot easier to take advantage of this infrastructure, the exchange of information is still mainly restricted to plain data, mostly in some kind of textual format. Multimedia (i.e., sound and image) data are slowly becoming more popular, with network capacity often being a bottleneck. One aspect that is often overlooked, however, is the possibility to exchange not only data, but also more complex *services.* These services could in particular be *methods* (i.e., implementations of algorithms) that have been made available to the public, together with some documentation of their input-output behavior. Remote users could then use the Web, possibly enhanced by an appropriate interface, to access these methods and to feed their own (or somebody else's) data into it.

The protocol underlying this process involves a number of sites, which in theory may all be different from each other, viz.,

- the location of the user;
- the original location of the method;
- the original location(s) of the data;
- the site where the computation is carried out.

There are numerous applications for such a *method base* both in research and in business. In research, it is both the *exchange* and the *verification* aspect that seem intriguing to an experimental scientist. While the exchange of methods between colleagues is already taking place (although not at the speed and comfort that seems possible with the network infrastructure currently available), the verification of results in the areas of applied computer science

and computational sciences is notoriously underdeveloped. We estimate that in experimental computer science, less than 10% of all results that have been published in scientific journals have ever been verified by other researchers. The tradition of reproducing experimental results, which is routinely performed in practically all other applied sciences (in particular medicine and engineering) simply does not exist in our own field. We believe that in the process of applied computer science maturing into an established discipline, this needs to be corrected. Modern computer networks provide the infrastructure for doing this, and tools like the one presented in this paper should help to use this infrastructure efficiently.

One area where method bases have been discussed for quite some time is decision support systems, and there in particular operations research. In those domains, method bases are often called *model management systems*, with the term *model* (as opposed to *method*) expressing that applying methods (such as a linear programming solver) is only one step in solving a complex optimization or decision problem. The application of the method(s) has to be preceded by the *design* of an appropriate model and succeeded by its *validation*.

After some papers in the late 70s and early 80s [DHL79] [MB79] [Bar80], Dolk [Dol86] presented a model management system for mathematical programming in 1986. Two years later, Jarke and Radermacher [JR88] published a paper on the role of model management in decision support. In the late 80s, several tools for algorithmic discrete mathematics were developed; see [DS94] for an overview. Mehlhorn and Näher, for example, presented their system LEDA, a portable library of data types and algorithms for computational geometry and combinatorial optimization [MN90]. Nievergelt and Schorn followed with their XYZ system in 1991 [NSA+91, Sch91]. These and other tools can be viewed as *small scale* model management where, based on a specific language and interface technology, researchers are supported in the rapid prototyping of their algorithmic research. Fourth-generation languages like Mathematica [Wol88] or Maple [CGG+88] follow a similar goal. A more recent approach was presented by Becker [Bec94], who is also the first author mentioning a possible interface of his system M to the World Wide Web. This approach heads for what could be called *large scale* model management, where the aim is high level integration of a heterogeneous world of model implementations. A similar goal is apparent in the work of Muhanna [Muh92]. The proposals for a megaprogramming language [WWC92] go into the same direction, although they focus more on the related database and software engineering aspects.

The WWW community has also started to use the Web as an interface to complex software tools. FitzGerald and Pearlstein [FP94], for example, show how to use WWW/Mosaic as a form-based frontend to a variety of applications in computational chemistry and computational molecular biology. A Web-based approach to provide a friendly interface for interactive visualization was described by Robertson et al. [RJN94]. Ibrahim [Ibr94] describes an innovative use of WWW clients as a tool for the teaching of algorithms and data structures. Related efforts include work on how to use WWW as an interface for the control of engineering tools [SHW94] or as a frontend to databases [Sjo94].

In the sequel, we will present *MMM*, a distributed model management system for *time series analysis and prediction*. Examples of time series range from the irregularity in a heartbeat to the volatility of a currency exchange rate. We focus on time series where the underlying deterministic equations are not known and where a model needs to be built from the data. Many disciplines, such as economics, econometrics, physics, biology, statistics, electrical engineering, medicine, have developed their specific methods of time series analysis. Collecting data and comparing methods from different disciplines has started only recently; the *Santa Fe Time Series Prediction and Analysis Competition* carried out in 1991 is one of the first examples [WG94].

The following section describes several scenarios where a tool like MMM could be useful. Section 3 explains why (and how) somebody working with time series should use MMM. In Section 4 we describe the architecture of MMM. Section 5 contains our conclusions and plans for the implementation phase.

2 Four Application Scenarios

In analogy to the terminology used in decision support system research, we use the term *model* for a specification of constrained, structured data, including the methods that can be applied to it. For example, a symmetric matrix is a model that is described by the dimension n of the matrix, and $n \times (n-1)/2$ rational numbers, each with a row and column index. An algorithm that inverts a symmetric matrix is a model with an input and an output, both symmetric matrices such that the output is the inverse of the input, and a method that computes the output from the input. A model management system is a software system that helps users to define, evaluate, combine

and compare such models [Muh92].

Scenario 1: Code Verification

After reading a paper about an interesting algorithm to invert symmetric matrices, a professor asks one of her students to implement the algorithm as a term project. A few days later, the student submits a C program with several hundred lines of code.

The first problem is how to check the code in order to give the student a fair grade. Secondly, we might want to make the program available to other people in the same research group. If the software had been written as part of a bigger project, the professor might have given the student detailed specifications in advance. But this was not the case here, so the functionality and the user interface are somewhat *ad hoc*. The third problem is to make the code robust towards incorrect initializations of the data structures (such as a non-symmetric matrix). Robustness is particularly important if we want to integrate the code into a larger software environment. Fourth, we might want to add some additional constraints (e.g., the matrix should have dimension 2 or more).

One approach that could solve most of these problems is to encapsulate the code in some kind of *container*. The normalized outer side of the container guarantees that the code's external interface is easy to understand and that it conforms to certain standards. This facilitates the communication not only with a human user but also with other software, such as a graphical user interface or a database system. The inner side of the container is organized such that it accepts only feasible initializations of the data structures. In our example this means that it can only contain symmetric matrices, possibly of dimension 2 or greater.

Providing software with special interfaces is done frequently. But it is usually expensive to do this and the result is not normalized. As a result, model management in the sense of combining different models on the fly is nearly impossible. We therefore argue for a *container generator* that, based on a simple specification language, produces the code to implement the container. A technique for container generation is described in Section 4.

Scenario 2: Experimental Design

This scenario concerns the testing of complex methods, possibly consisting of a variety of hierarchically structured submodules. For example, a method may involve different levels of partial computations, which are triggered by the selection of parameters. This means in particular that, depending on previous computations, different methods may be feasible or appropriate at various stages. Typical user interfaces in such cases are hierarchically organized collections of forms (or menus) where one can navigate at random, setting parameters and applying submethods until one obtains a result. An example is the system MulTi for multivariate time series analysis [HLC+92].

One problem in this scenario is that of keeping a history of parameter settings, i.e., of the exact way a particular experiment has been performed. Besides being able of repeating the same experiment at another time, this is also crucial for supporting systematic tests of the impact of certain parameters. Another question is how to publish the results such that the experiment can be understood by the research community.

The requirements on a model management system implied by this scenario are a technology that represents such a hierarchical method collection as an instance of one complex model. All data as well as parameter initializations should be part of the model, such that storing instances means keeping a history of experiments. For the container system described in the first scenario, this means that one requires large containers that combine small containers in a structured way. The normalized outer sides of the small containers are used to organize their cooperation inside the large container.

Scenario 3: Choosing an Appropriate Method

Let us consider a particular mathematical operation, such as the computation of eigenvalues of quadratic matrices. There is a large number of methods to solve this problem, and it would indeed be attractive to have some kind of *method selection tool* that knows about the strengths and weaknesses of the various algorithms and that can choose one that seems most appropriate for a given input matrix.

In statistical applications, this problem is even more relevant. Here, a main part of the problem is to choose an appropriate model, e.g., a linear versus a non-linear system in time series analysis. Otherwise the result of the method (e.g. the estimators for the model parameters) may be unsatisfactory, simply because the model does not provide an appropriate fit for the given data.

The main question is how to obtain and represent the required knowledge about the methods and models involved. One approach is to interview experts and represent their opinions symbolically, using tra-

ditional knowledge representation techniques. This expert system approach has been tried with varying degrees of success. A more empirical approach, which seems promising to us, is to start with a model management tool that does not have a selection component yet, i.e., users have to chose their preferred methods themselves. One can then *monitor* how methods are being used and how people apply certain methods to a given problem or problem class. Once a sufficient amount of this *metadata* has been collected, it can be used to guide other users in their search for an appropriate method.

Scenario 4: Verifying Research Results

Assume you have to referee a research paper describing a new method. The paper reports excellent results for a certain class of data instances. The problem is how to judge the results without implementing the method. How can the data be accessed, such that other methods can be run against it? How can the methods be run against your own data? Or how can you at least repeat the authors' experiments on your own workstation?

The solution is a model management system that supports the access to method implementations of other research groups. This support might be given by integrating their code into other environments, or by invoking the methods via the Internet. The latter variant requires that authors of methods are provided with a technology to make their methods accessible in a normalized manner.

3 MMM and Time Series Analysis

In this section we explain why we chose time series prediction and analysis as the domain for MMM, why somebody might be interested in using methods and data that have been contributed to MMM, and why a time series researcher might want to contribute anything to MMM in the first place. We also briefly describe which methods are available in the current prototype. The section concludes with thoughts on the collection of metadata, i.e., an automated analysis of the use of MMM that could yield insights into relations between data and algorithms.

3.1 Time Series Analysis and Prediction

Time series analysis has three goals: forecasting, modeling, and characterization. The aim of forecasting is to accurately predict the short-term evolution of the system; the goal of modeling is to find a description that captures features of the long-term behavior of the system. These are not necessarily identical: finding governing equations with proper long-term properties may not be the most reliable way to determine parameters for good short-term forecasts, and a model that is useful for short-term forecasts may have incorrect long-term properties. The third goal, system characterization, attempts with little or no a priori knowledge to determine fundamental properties, such as the number of degrees of freedom of a system or the amount of randomness. This overlaps with forecasting but can differ: the complexity of a model useful for forecasting may not be related to the actual complexity of the system.

It is useful to characterize the complexity of a model on an axis that ranges from strong models to weak models. Strong models have strong assumptions. They are usually expressed in a few equations with a few parameters, and can often explain a plethora of phenomena. In weak models, on the other hand, there are only a few domain-specific assumptions. To compensate for the lack of explicit knowledge, weak models usually contain many more parameters (which can make a clear interpretation difficult). It can be helpful to conceptualize models in the two-dimensional space spanned by the axes data-poor↔data-rich and theory-poor↔theory-rich. Due to the dramatic expansion of the capability for automatic data acquisition and processing, it is increasingly feasible to venture into the theory-poor and data-rich domain.

Two crucial developments occurred in the last decade [Wei94]; both were aided by the general availability of powerful computers that permitted much longer time series to be recorded, more complex algorithms to be applied to them, and the data and results of these algorithms to be visualized interactively. The first development, state-space reconstruction by time-delay embedding, drew on ideas from differential topology and dynamical systems to provide a technique for recognizing when a time series has been generated by deterministic governing equations and, if so, for understanding the geometrical structure underlying the observed behavior [SYC91]. The second development was the emergence of the field of machine learning, typified by neural networks, that can

adaptively explore a large space of potential models.[1]

One of the first examples in history where a number of methods was simultaneously applied to a number of data sets was the *Santa Fe Time Series Prediction and Analysis Competition*, carried out in 1991 and followed up by a NATO workshop in 1992 [WG94]. The competition focused on six data sets, ranging from 1,000 to 100,000 points. All of the successful entries were fundamentally nonlinear and, even though significantly more computer power was used to analyze the larger data sets with more complex models, the application of the techniques required more careful manual control than in the past. There was a general failure of simplistic "black-box" approaches. In all successful entries, exploratory data analysis preceded the application of the algorithm. The Santa Fe competition showed examples of nonlinear results going far beyond what is possible within the canon of linear systems analysis, but also showed that there are unprecedented opportunities for the analysis to go astray.

Scientific integrity in the competition was enforced by withholding the continuations of the data sets until after the deadline. It would have been preferable, however, if a framework had been available where participants check-in their models and the judges test them at their convenience. Among other things, this would have allowed statistical tests with much more data than could possibly be asked for at the time of submission.

But not only the evaluation in a competitive setting is crucial. Techniques for the analysis and prediction of time series have been developed by a number of disciplines, such as physics, econometrics, biology, statistics, electrical engineering, medicine, and machine learning—but rarely do people in one field know about the methods of the other fields. Our MMM system provides an infrastructure that allows approaches from these traditionally disjoint disciplines to come together. This enables researchers in one field to explore state-of-the-art methods from other fields - another task that has been impossible up to now.

So far we have considered users equipped with data who are in search of analysis methods. Now we turn to researchers who develop methods and who are in search of data to prove that their methods are good. Up until a few years ago, it was not uncommon for scientists to have their data printed in an appendix to their paper. If other researchers wanted to try their algorithm on the same data set and maybe to compare algorithms, they often just typed that data back in. Similarly, people reimplemented algorithms that other people had published in plain English in order to analyze data they were interested in.

With the increase in data set size and in the complexity of the algorithms (in particular there are often many parameters that have to be set from experience rather than from first principles), this approach has become prohibitive. Also, with the shift from simple, restrictive models to more complicated analysis methods it has become important to analyze data with several methods, and methods with several data sets. We hope that MMM will serve as playground that will facilitate the exploration of both data and algorithms. In the present prototype version of MMM the following methods have been included.

- **Visualization.** We embedded XGobi, a tool developed at AT&T Bell Labs for the visualization of three (or more) dimensional data [SCB92].

- **Subset Selection.** When confronted with a new set of data, one of the first questions is: Which of the variables that could potentially serve as inputs contain information about the output? We have implemented the method by Bonnlander and Weigend [BW94] to find the subset of variables (or *input features*) that has maximal mutual information with the output (i.e., the value to be predicted).

- **Redundancy.** A relatively recent development for the characterization of time series has been the use of incremental mutual information, also called redundancy: it measures how many additional bits of information we learn about the next value when we add another lag to the input. We adopted the code used by Gershenfeld and Weigend [WG94].

- **SNP.** SNP by Gallant and Tauchen [GT94] is a method for non-parametric time series analysis that employs a polynomial series expansion to approximate the conditional density of a multivariate process.

3.2 The Role of Technology

We see several levels of impact that technology has made on the times series analysis and prediction domain. Commonly, the first and the third of these levels are seen as infrastructure, but we believe that it is important to realize to what degree technology is

[1] There is a great amount of literature on this topic; a good recent volume is edited by Smolensky, Mozer and Rumelhart (1995)[SMR94], as well as the annual conference proceedings of *Advances in Neural Information Processing Systems*, e.g., 1995 [NIP95].

changing both the techniques and also the way people are working on time series.

- **Collection of Data and Algorithms.** Traditionally, this has been viewed as the central part of research work; collecting data was (and in many areas still is) very hard, and the implementation of algorithms and models is only slowly getting easier through more friendly interfaces.

- **Exploration of Data and Algorithms.** The explorative analysis, going beyond stock items like visualization and sonification, is becoming an integral part of scientific inquiry. In the long run, we view this less as a workbench (with a usually well defined set of tools) than - due to the huge number of degrees of freedom - as a playground, an interactive media space for the exploration of data, models and their relations. This is where we expect the main leverage of MMM to be.

- **Communication.** By sharing the Internet and WWW as common workspace, the time series community as a whole has started to do things differently. We view our work as providing easy access to other people's tools in this large workspace.

- **Metadata.** Researchers today are often faced with the choice of either using a well understood technique (such as a linear model) that is too restrictive and inappropriate, or a broader technique which not much can be proven about. The methods at present have become so complex that an explicit analysis is not possible. Their use, and the experiences of the users, take the place of explicit analysis. One important motivation for MMM thus is to collect metadata about the use of the data and algorithms in order to gain insight into patterns of more (or less) successful applications. We see a stepwise approach to reach this goal. The first step is to put up a system to which researchers can easily submit their methods. The second is to allow users of the system to make annotations to services that they enlisted and to make these public to other users. The third step is to go from informal to formal annotations that can be evaluated by computer programs.

An interesting new development where several of these issues are touched is the time series WWW site that has recently been created at the University of Colorado at Boulder. Contributing data sets, papers, algorithms and annotations is easy: All it takes is filling out a short form (`http://www.cs.colorado.edu/Time-Series/Submit.html`). That time series site also serves as a testbed for the *Harvest Information Discovery and Access System* [BDH+94], which efficiently gathers and indexes the Santa Fe data using corpus-specific customizations. But even in this advanced application the exchange of information is still restricted to obtaining data (time series data and their visualizations and sonifications), papers, and code, as well as discovering papers that use these data.

4 The Architecture of MMM

Following the idea of containers discussed in Section 2, MMM is implemented as a system of abstract data types that provide a normalized integration of all desired functionalities. The abstract data types are combinations of data and methods; they are implemented as object-oriented class libraries. The data types encapsulate or implement directly

1. implementations of application models, such as Gauss or Matlab scripts, or routines written in FORTRAN, Pascal, C or C++;

2. model management functionalities;

3. metamodels consisting of high-level descriptions of the models of types 1 and 2.

We have chosen the C++ code generator of the *Ypsilon* system [KLMM94] to generate the container classes. Ypsilon generates *safe* data structures (in the sense of [MM94]). Such data structures implement the structure of a model as well as the constraints that define feasible initializations of the model. Classes generated by Ypsilon have a normalized set of methods for basic functionalities such as *create, delete, edit, evaluate, read,* and *write.* Ypsilon encapsulates methods in *function classes*, which each have an input field, an output field, and a method called *evaluate* to call the routine for computing the output from the input. This design is similar to what is called operational programming in [MS94, Sol87] or *megaprogramming* in [WWC92].

The abstraction allows to implement meta algorithms on objects of function classes that are independent of the specific type of the functions. For example, the current version of Ypsilon comes with the realization of various *Eigenmodels* [BMR89, MMR94], i.e., special models that represent the structure of

parts of the system. The realized Eigenmodels are variations of *data flow diagrams*, in which the user is provided a graphical tool to express interoperability. They implement an initial set of meta models of MMM functionalities.

The MMM system is designed as a client/server architecture, consisting of method servers, a method agent and clients that communicate with the method agent in order to access a method server (see Figure 1). We explain these components along two typical MMM operations: (i) contribution of a new method to MMM, and (ii) access to a method that is located on some method server.

There is no doubt that participation in a method base such as MMM will place an additional burden on the implementor of a method. In order to make a method publicly available *(method integration* or *method check-in)*, one has to

- select the modules to be made available (in the case of a complex method, one may choose to provide only partial functionality, given the complexity and licensing requirements of certain submodules);
- describe the input-output behavior of the selected modules, possibly in some standard description language;
- define the formats of input and output files.

In MMM this is basically done by describing an interface to the method in terms of Ypsilon classes. Three classes have to be identified:

1. an input class;
2. an output class;
3. a function class.

The *evaluate* method of the function class has to be filled by the call of the method, including a data conversion to and from the method-specific data formats, if necessary. The method check-in is illustrated in Figure 2.

The method provider generates implementations from the descriptions of the Ypsilon classes, compiles these and loads them to an Ypsilon method server. This server is a simple pipe that reads Ypsilon objects, initializes an object of the function class, invokes *evaluate*, and returns the result via standard output. The server is invoked by a CGI program [McC94] that can be accessed via the Internet by sending HTML forms to an http daemon. Figure 3 shows the architecture of an Ypsilon method server.

The descriptions are propagated to the method agent, which stores them as part of its metadata. It makes them accessible to other users by compiling and loading them to the method agent. An important technical detail is that the method agent implements a function class in which the original code of the *evaluate* method (calling some methods) is replaced by a communication with the method server. If the method agent receives a request from a client to invoke the *evaluate* method of a function class, it contacts the method server, sends it the instance of the object, and gets back the result of the *evaluate*.

The client is a program that communicates with the method agent by sending commands or editing objects. Commands include creating a function object, initializing its input with data, or connecting to a method server. The user interface of MMM is based on HTML forms and documents. This means in particular that WWW clients can be used for communication with the method agent (possibly with a CGI program in between). Building on HTML also allows to weave in references to documents available on the Internet. A typical Ypsilon class description, for example, can then easily be enhanced by pointers (URLs) to related documentation. Finally, by basing the user interaction on HTML, any WWW client will work as an interface to the method agent.

5 Conclusions

This paper describes the basic design of MMM, a model management system for time series analysis. The long-term goal of the MMM project is to offer a WWW-based distributed infrastructure for interdisciplinary method exchange and model management. MMM is based on several recent developments in time series analysis, model management systems and information systems. A particularly important goal of MMM is to make algorithmic research results available to a larger audience and thus thoroughly verifiable by the research community.

A simple first prototype with the time series methods listed in Section 3 has recently been implemented [MRS94] and can be tested through the World Wide Web
under URL `http://ischtar.wiwi.hu-berlin.de/many_mmi/many_mmi.html`. For the next phase, we intend to extend the method agent to act as a relay and conversion point for data. Model instances could then be stored with the method agent, so the user does not have to use an ftp server to provide the method with data. Furthermore, the method server can send results directly to the method agent, where

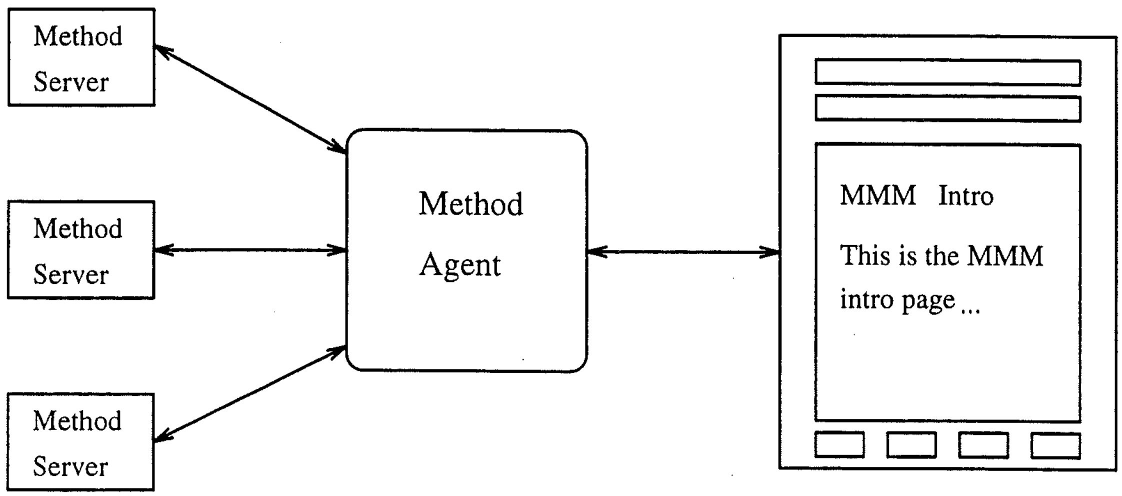


Figure 1: MMM Architecture

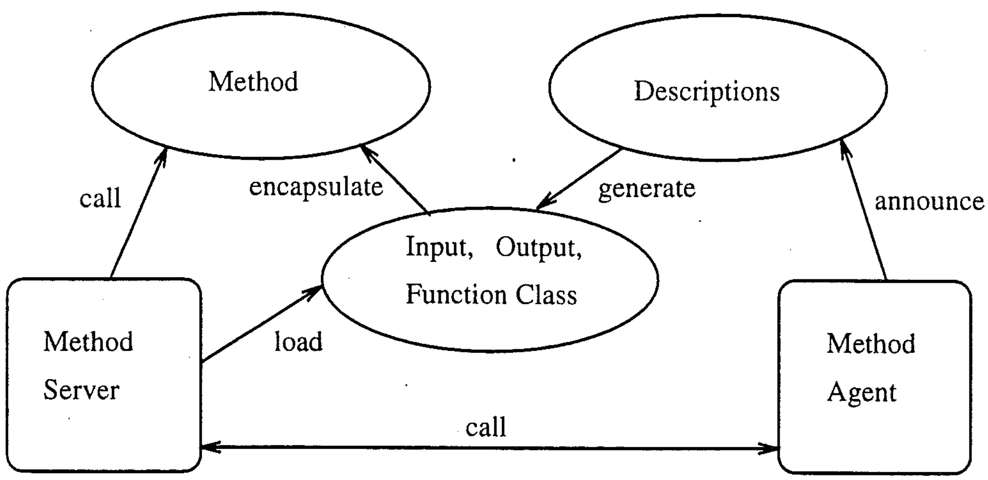


Figure 2: Check-In of Methods

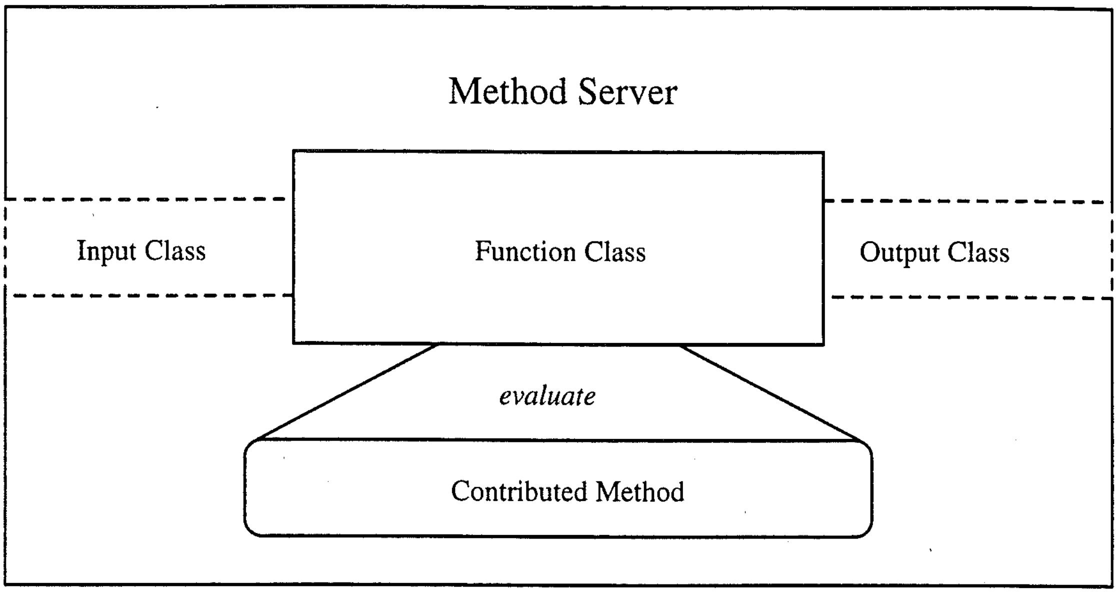


Figure 3: Ypsilon Method Server

they can be viewed by the user as an HTML document.

Furthermore, the method agent should soon serve as a gatherer of metadata that is accessible by method providers and method users. In the first stage, the metadata are all integrated function models as well as URLs of related information. Based on this metadata, the method agent can support the check-in of new methods. One possible functionality would be to use keywords and mathematical subject classification indices in order to find a set of descriptions for modeling the interface to new methods. Research groups may then agree on a basic set of data models to which they connect their methods.

Acknowledgments

The authors acknowledge support from the Deutsche Forschungsgemeinschaft under the Sonderforschungsbereich 373. Andreas Weigend also acknowledges support from the National Science Foundation under Grant No. RIA ECS-9309786.

References

[Bar80] H. Barth. Grundlegende Konzepte von Methoden- und Modellbanksystemen. *Angewandte Informatik*, 8:301–309, 1980.

[BDH+94] C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz. The Harvest information discovery and access system. In *Proceedings of the Second International World Wide Web Conference.* `http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Searching/schwartz.html`, 1994.

[Bec94] P. Becker. M - An object-oriented model and method base system for discrete optimization. In *Proceedings of the International Conference on Object Oriented Information Systems (OOIS'94)*, London, UK, 1994.

[BMR89] M. Bartusch, R. H. Möhring, and F. J. Radermacher. Design aspects of an advanced DSS for scheduling problems in civil engineering. *Decision Support Systems*, 5:312–344, 1989.

[BW94] B. V. Bonnlander and A. S. Weigend. Selecting input variables using kernel density estimation. In *Proceedings of the 1994 International Symposium on Artificial Neural Networks (ISANN '94)*. Tainan, Taiwan, 1994.

[CGG+88] B. W. Char, K. O. Geddes, G. H. Gonnet, M. B. Monagan, and S. M. Watt. *Maple Reference Manual.* WATCOM Press, 1988.

[DHL79] K. R. Dittrich, R. Hübner, and P. C. Lockemann. Methodenbanksysteme: Ein Werkzeug zum Maßschneidern von Anwendersoftware. *Informatik-Spektrum*, 2:194–203, 1979.

[Dol86] D. R. Dolk. A generalized model management system for mathematical programming. *ACM Transactions on Mathematical Software*, 12(2):92–126, 1986.

[DS94] N. Dean and G. Shannon, editors. *Computational Support for Discrete Mathematics - DIMACS Workshop, March 12-14, 1992.* American Mathematical Society, 1994.

[FP94] P. C. FitzGerald and R. A. Pearlstein. The Web as a computational engine for chemistry and molecular biology. In *Proceedings of the Second International World Wide Web Conference.* `http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/BioChem/pearlstein/fitzgerald.html`, 1994.

[GT94] A. R. Gallant and G. Tauchen. SNP: a program for nonparametric time series analysis, Version 8.3, Users Guide. Technical report, Department of Economics, University of North Carolina, 1994.

[HLC+92] K. Haase, H. Lütkepohl, H. Clausen, M. Moryson, and W. Schneider. MulTi - a menu-driven GAUSS program for multiple time series analysis. Technical report, Institut für Statistik und Ökonometrie, Universität Kiel, 1992.

[Ibr94] B. Ibrahim. World wide algorithm animation. In *Proceedings of the First International World Wide Web Conference.* `http://www1.cern.ch/PapersWWW94/bertrand.ps`, 1994.

[JR88] M. Jarke and F. J. Radermacher. The AI potential of model management and its central role in decision support. *Decision Support Systems*, 4:387–404, 1988.

[KLMM94] D. Kühl, A. Ludwig, R. H. Möhring, and R. Müller. *Ypsilon User Manual.* Technische Universität Berlin, 1994.

[MB79] P. Mertens and F. Bodendorf. Interaktiv nutzbare Methodenbanken - Entwurfkriterien und Stand der Verwirklichung. *Angewandte Informatik*, 7:533–541, 1979.

[McC94] Rob McCool. *The Common Gateway Interface.* `http://hoohoo.ncsa.uiuc.edu/cgi/overview.html`, 1994.

[MM94] D. Möller and R. Müller. A concept for the representation of data and algorithms. In N. Dean and G. Shannon, editors, *Computational Support for Discrete Mathematics, DIMACS Workshop March 12-14, 1992.* AMS, 1994.

[MMR94] R. H. Möhring, R. Müller, and F. J. Radermacher. Advanced DSS for scheduling: Software engineering aspects and the role of Eigenmodels. In J. F. Nunamaker and R. H. Sprague, editors, *Proceedings of the 27th Annual Hawaii International Conference on System Sciences*, volume III, 1994.

[MN90] K. Mehlhorn and S. Näher. LEDA, a library of efficient data types and algorithms. In M. Nagl, editor, *Graph-Theoretic Concepts of Computer Science*, volume 411 of *Lecture Notes in Computer Science*, pages 88–106. Springer, 1990.

[MRS94] R. Müller, W. B. Rubenstein, and P. Schmidt. A simple method server for the Web. Working paper, Sonderforschungsbereich 373, 1994.

[MS94] R. Müller and D. Solte. How to make OR results available: a proposal for project scheduling. In W. Gaul, F. J. Radermacher, and D. Solte, editors, *Data, Expert Knowledge and Decisions*, Annals of Operations Research. J.C. Baltzer Science Publishers, 1994. to appear.

[Muh92] W. A. Muhanna. On the organization of large shared model bases. *Annals of Operations Research*, 38:359–396, 1992.

[NIP95] *Advances in Neural Information Processing Systems 7 (NIPS '94).* Morgan Kaufmann, San Francisco, CA, 1995.

[NSA+91] J. Nievergelt, P. Schorn, C. Ammann, A. Brünger, and M. De Lorenzi. XYZ: A project in experimental geometric computation. volume 553 of *Lecture Notes in Computer Science*, pages 171–186. Springer, 1991.

[RJN94] D. Robertson, W. Johnston, and W. Nip. Virtual frog dissection: Interactive 3d graphics via the Web. In *Proceedings of the Second International World Wide Web Conference.* `http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/BioChem/robertson/robertson.html`, 1994.

[SCB92] D. F. Swayne, D. Cook, and A. Buja. User's Manual for XGobi, a Dynamic Program for Data Analysis Implemented in the X Window System (release 2). Technical memorandum, Bellcore, 1992.

[Sch91] P. Schorn. Implementing the XYZ GeoBench: A programming environment for geometric algorithms. volume 553 of *Lecture Notes in Computer Science*, pages 187–202. Springer, 1991.

[SHW94] R. Scharf, S. Hartmann, and W. Wolz. Using mosaic for remote test system control supports distributed engineering. In *Proceedings of the Second International World Wide Web Conference.* `http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/CSCW/scharf/scharf.html`, 1994.

[Sjo94] M. Sjolin. A WWW front end to an OODBMS. In *Proceedings of the Second International World Wide Web Conference.* `http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Databases/sjolin/sjolin.html`, 1994.

[SMR94] P. Smolensky, M. C. Mozer, and D. E. Rumelhart, editors. *Mathematical Perspectives on Neural Networks.* Erlbaum Associates, Hillsdale, NJ, 1994.

[Sol87] D. Solte. *Open Systems - Ein lernendes Verwaltungssystem für die rechnerunterstützte Methodenkonstruktion im Bereich des Operations Research*, volume 38 of *VDI-Forschungsberichte, Reihe 16.* VDI-Verlag, 1987.

[SYC91] T. Sauer, J. A. Yorke, and M. Casdagli. Embedology. *Journal of Statistical Physics*, 65:579–616, 1991.

[Wei94] A. S. Weigend. Paradigm change in prediction. *Philosophical Transactions of the Royal Society (Physical Sciences)*, page 348, 1994.

[WG94] A. S. Weigend and N. Gershenfeld, editors. *Time Series Prediction.* Addison-Wesley, 1994.

[Wol88] S. Wolfram. *Mathematica - A System for Doing Mathematics by Computer.* Addison-Wesley, 1988.

[WWC92] G. Wiederhold, P. Wegner, and S. Ceri. Toward megaprogramming. *Communications of the ACM*, 35(11):89–99, 1992.

Multimedia Information Delivery and the MHEG Standard

Chetan Gopal, Roger Price*
Distributed Multimedia Systems Laboratory
Department of Computer Science
University of Massachusetts Lowell
e-mail: {cgopal, rprice}@cs.uml.edu

Abstract

MHEG is an ISO committee effort that is currently at the Draft International Standard (DIS) stage. The MHEG standard defines a coded representation of hypermedia objects intended for final-form presentation and real-time interchange over wide area networks and is intended to be a basis for major industrial developments. We present the MHEG design for supporting real-time delivery of information multimedia objects.

1 History

The work on MHEG began in ISO in 1989 as an adhoc group led by Mr. Francis Kretz of France Telecom, following an initiative by Dr. Hiroshi Yasuda of NTT. The committee is now ISO working group JTC1/SC29/WG12, also known as "MHEG". This group is a part of the well known "JPEG, JBIG, MPEG" activity of SC29. The specification of the Draft International Standard is currently being prepared, and has entered the formal balloting process.

Although MHEG is not an architecture, it has been influenced by the earlier experimental French standard RAVI and the Japanese work on Tron. In particular, the RAVI architecture aimed at providing a basis for a very wide area deployment of interactive audio-visual applications across multiple networks, using heterogeneous equipment. RAVI was successfully demonstrated in 1988, and in 1989 a transatlantic demonstration showed attendees at a conference at an IBM facility in Thornwood NY several interactive applications running from servers in France. These applications included audio, photographic images, graphics, text and user interaction. One of the lessons learnt from RAVI is the need to support independent industries for the creation of interactive material, the operation of the servers, and the operation of the networks. The French experience was greatly facilitated, as is the hugely successful Minitel by a fully integrated billing system.

Several reviews of MHEG have appeared previously [3][4]. This paper is based on the most recent version of the draft DIS, and also reviews recent implementation work.

2 Introduction

In recent years interest in multimedia applications and information delivery has increased dramatically. Also, there is a growing desire among content developers, application developers and end users to easily reuse and interchange multimedia information across heterogenous platforms. It is not sufficient to interchange data; one must also be able to interchange structural, spatial and temporal information related to the composition of the multimedia objects. The MHEG standard is not aimed at any one application, it is intended to help federate a wide range of applications. Neither is it a restrictive, prescriptive standard, rather it is intended as an enabler for other international, industrial and proprietary standards. Accordingly, four modes of interchange have been identified[6][3]:

1. As a final storage model during the creation and editing of multimedia documents, for both the new composition and for archival materials which may be used in the editing process.

2. As a format for delivery of final-form digital media, for example, by compact disks, to end-user players.

3. As a format real-time delivery from a server to clients connected via a network for training, information-on-demand, etc.

*Currently on leave from IBM France.

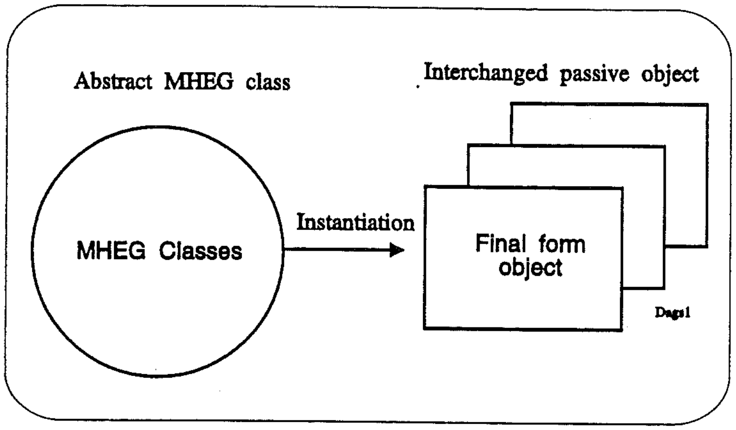


Figure 1: Scope of MHEG.

4. For inter-application exchange of data.

With this in view the ISO committee is designing a standard for multimedia interchange representation. The proposed MHEG standard can be compared informally with the ITU Recommendation T.4 for facsimile exchange known popularly as Group 3 fax. This highly successful de jure standard provides a detailed description of a coded representation of the appearance of a page of information, be it typed, printed, hand-written or generated by a computer. Rec T.4 provides an agreed coded representation for the interchange, but does not say how that encoding is to be produced, neither does it say what is to be done with the encoding received. This is left to the implementer, and has led to a wide choice of facsimile equipment to meet every taste and budget.

Similarly, MHEG provides a specification for an encoded representation of a set of hypermedia objects. These objects describe a generic behavior, frequently an interaction between a user and the presentation system, but the standard does not say how this is to be implemented. Clearly the products of different manufacturers will have their own "look and feel".

The term "MHEG" is used both for the ISO working group which is developing the standard, the standard itself, and the objects it defines. Its usually clear from the context what is meant. The acronym is usually pronounced as two syllables in English and one syllable in French, the working languages of the ISO.

The standard has three parts: Part 1 deals with ASN.1[8][9] as the base encoding notation, Part 2 deals with SGML encoding and Part 3 provides for MHEG extensions for scripting language support. The following sections relate to part 1 of the standard alone. Parts 2 and 3 are still in the process of development and too immature to be detailed.

An MHEG FAQ is available at:

`ftp://devo.uml.edu/pub/MHEG/MHEG.FAQ`

3 Scope of MHEG

The scope of MHEG is limited to coded representation of final form multimedia and hypermedia information objects and their interchange in the four modes mentioned in section 2: The standard requires that all interchanged objects be conformant to the entire standard i.e. they can be correctly interpreted, and does not specify any conformance with respect to the MHEG engine. There are no options. This eliminates all the problems that arise with incompatible subsetting. MHEG has adopted an object oriented approach to specify MHEG object classes, i.e. it supports aggregation, inheritance and polymorphism. MHEG objects, their instantiation and interchange are what is within the scope of the standard. In addition to this, MHEG defines generic behaviors on active instantiation of MHEG objects (known as run-time objects) which every MHEG engine has to conform to. There is no formal MHEG engine defined by this standard.

3.1 Object Classes [1][2]

MHEG defines a set of classes, and objects (MHEG objects) are instantiations of these classes. These

MHEG objects are considered to be passive information entities rather than active entities. These interchangeable MHEG objects have no relevance to objects in any other object oriented paradigm where objects tend to be active. Once MHEG objects have been interchanged, active objects may be instantiated from these passive objects within the processing MHEG engine. These active objects are referred to as run-time objects in MHEG terminology. MHEG objects provide for the following functionality:

1. Support for final form presentation: Sufficient generic presentation information can be encoded into the MHEG objects so the MHEG engine can present them with out restructuring. It is up the MHEG engine to map these presentation requirements on to the capabilities of the underlying platform and its multimedia system service capabilities.

2. Support for minimal resources: Since MHEG is aimed at minimal systems, all objects contain information which help minimal systems handle these objects fairly well. Because of this flexibility MHEG does not guarantee isomorphic presentations across platforms.

3. Support for interaction and synchronization: One important feature of MHEG is that it provides facilities to describe results of user interactions rather than pro- vide interaction facilities. All interaction behaviors are based on generic selection and modification behaviors of run-time components and sockets. MHEG provides for all four synchronization classes identified, namely, elementary, chained, cyclic and conditional synchronizations through the use of the link object. The conditional synchronization will be described in more detail later in this paper.

4. Support for real-time presentation and interchange: The intent of MHEG is to provide for real-time interchange, but MHEG makes no assumptions about the underlying system services' real-time capability. With this in mind MHEG provides real-time support in two ways:

    - Real-time requirements can be satisfied by defining a set of object specifications.

    - The descriptor object provides capabilities to define real-time requirements through the use of Quality of Service specification.

3.2 Sub Classes and Inheritance

MHEG classes are defined in such a way that classes are derived from or inherit properties of other classes. For example, Container class inherits properties from Link class to represent any start-up links. This clarifies the standard, but does not require that an implementation be based on object oriented techniques.

3.3 Polymorphism

MHEG objects when active, can be acted upon by the MHEG engine with certain actions. These actions are also applicable to the active object's subclasses. The effect of these actions are not isomorphic. For example, the *RUN* action on content objects like an image and an audio clip can result in two entirely different presentations.

4 MHEG and Delivery Architectures

MHEG fits well into a number of delivery architectures. This section will exemplify some of such architectures where MHEG will fit in. Examples described below will use the same functional MHEG model to ease understanding. A brief description of the components of the model are given below.

- The MHEG communication module will handle all networking and communication of the MHEG engine with the outside world

- The MHEG engine which processes and schedules objects for presentation and interchange.

- An user interface through which an user can interact with MHEG objects.

- A user application which communicates with the MHEG engine through a given set of API defined by the engine implementation. It also receives interaction and status events from the user interface and the MHEG engine respectively.

Please note that these are examples: the implementer is free to choose other models.

4.1 Interactive Telematic Service Model

Figure 2 shows possible actions from a user terminal which is in session with a main server. The user terminal interacts with the main server requesting both

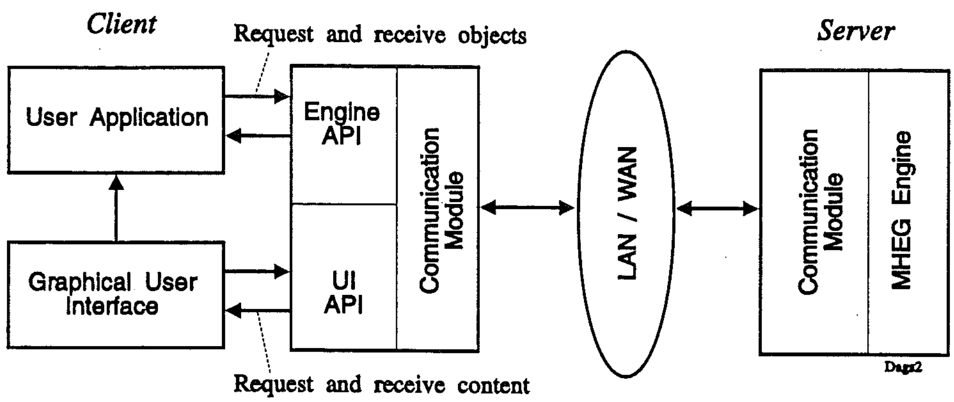


Figure 2: Interactive Telematic Service.

MHEG objects and media content. This is very similar to a WWW application like Mosaic[1] communicating with an HTML server, or a gopher client retrieving files and menus from a Gopher server. The user application and its interface are not defined by the MHEG standard, but are subjects of ongoing study in the ITU in the T.17X series of recommendations.

4.2 A Computer Supported Collaborative Work (CSCW) Model

Since MHEG provides support for real-time interchange, it is ideally suited for CSCW. See figure 3. Here user applications can author and modify MHEG objects at the same time. In this model the MHEG engines communicate with each other through the communication module. User at site-A modifies an object, for example, an architectural layout of a house. This object is encoded into ASN.1 in real-time and transmitted over the network to the MHEG engine in site-B where it is presented to the user. The user at site-B can also modify the same object and send it back and forth.

5 Relationship with HyTime

ISO 10744:1992 "Hypermedia Time-based Structuring Language (HyTime)" has recently been published as an international standard and potential users frequently ask why "further work" in this area is needed. HyTime and MHEG are two different species designed to work in different contexts. The HyTime standard makes no assumptions about the nature of its users. The MHEG standard assumes that its users are "industrial strength" interactive applications based, for example, on wide area digital networks. The techniques used are different: for example, the HyTime link is a very powerful and general form typically requiring further processing, whereas the MHEG link is in a "final form" requiring no change of structure.

A basic difference between a HyTime engine and an MHEG engine is that in general a HyTime engine sees the whole of a document or a script before it is executed, but an MHEG engine might often see only that small part of the document that the user works with, during the interaction.

6 Some Key Technical Features

We now discuss some of the technical features of MHEG which make it particularly suited to future network based "industrial strength" audio-visual applications.

6.1 Generic Space

The hypermedia objects and their behavior require an agreed representation of time and space, but this is not based on any particular presentation system. The space used is a generic simplification known as "Generic space". It provides for 3 spatial axes known

[1]NCSA Mosaic was developed at the National Center for Supercomputing Applications at the University of Illinois in Urbana-Champaign.

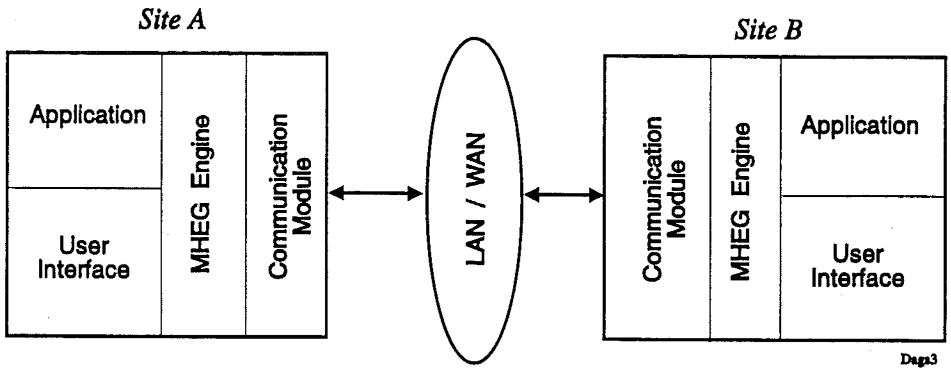


Figure 3: Computer Supported Cooperative Work.

as X, Y and Z, which are each defined on a finite interval $[-32768, 32767]$. The interval can be changed, but authors are encouraged to stay with the default axes. The presentation system is responsible for mapping this into the real world of a screen or window in a graphical user interface.

There is also a single time axis T defined on the interval $[0, \infty)$. The default unit is the millisecond, and authors[2] are encouraged not to change this.

6.2 Addressing

The MHEG standard provides a range of addressing techniques for objects. The standard distinguishes clearly between the way in which an address is defined, and the way in which the address is referenced.

6.2.1 External addressing

These addresses are defined outside of the MHEG standard by using the international standard IS 9070, which provides techniques for registering the owners of address spaces, and referring to addresses in this spaces. These techniques also support references to "system" addresses such as file names and Uniform Resource Locators (URL)[3].

MHEG shares this external addressing method with Apple's Bento.

For example:

```
-//IETF RFC 1630//
 ftp://devo.cs.uml.edu/MHEG/welcome.mhg
```

[2]The term authors may mean a human, but in general will be some computer based system offering a friendly interface to an audio-visual specialist.

[3]See RFC 1630 "Universal Resource Identifiers in WWW"

6.2.2 Internal addressing

This mode of addressing is defined in MHEG, and uses sequences of integers. The standard distinguishes between the addresses of the interchanged MHEG objects, and the addresses of the "runtime" objects known as rt-objects manipulated by an MHEG engine during a presentation.

1. An MHEG identifier for an interchanged MHEG object is made up of a sequence of integers $a_1, a_2, \ldots$ which identify the application, followed by a unique integer i identifying the object within the application:

   $$a_1.a_2 \ldots a_n.i$$

   It is the author's responsibility to ensure that the object number is unique.

2. An rt-object identifier is an integer specified by the author which is unique to the application. Re-use of an existing number may overwrite the previous object. A reference to an rt-object is made up of any form of reference to an MHEG object, followed by the integer indentifying the rt-object created from the original MHEG "model" object.

6.2.3 Symbolic addressing

An author may define alphanumeric aliases for any reference to an MHEG object, rt-object or any other object. In order to facilitate the binding of MHEG objects and rt-objects to scripts, it is suggested that authors limit the sophistication of their identifiers to that of the SGML reference concrete syntax.

```
alias START
      -//IETF RFC 1630//
       ftp://devo.cs.uml.edu/
           MHEG/welcome.mhg
```

6.3 Links and Actions

These two objects provide the mechanism for the description of a general generic behavior. The behavior is typically that of a simple "knee-jerk" reaction to a status change. More sophisticated logic requires the use of a script not defined by MHEG part 1.

Each link contains one or more triggers, followed by a set of actions. The triggers contain two elements, the trigger itself, and additional conditions which are to be satisfied if the link is to fire. The additional conditions may be thought of as a "safety catch". The trigger is made up of two predicates: *previous condition* and *current condition*. Intuitively, when an object passes from *previous condition* to *current condition*, the link is fired. For example the previous condition may be *rt-object was not running* and the current condition may be *rt-object is running*. A link which fires on this trigger is able to synchronize other presentations with the start of this one.

A typical MHEG engine might place all the links in a link processor which surveys the specified conditions and signals the MHEG engine when a link fires.

The actions form a recursive structure which is capable of considerable complexity, but in general may be thought of as directing a sequence of the primitive actions defined by the MHEG standard to a target rt-object. Typically such actions would create the presentable object, initialize its parameters, such as volume and position, and start its presentation.

6.4 Conditional Synchronization

Clearly MHEG should be capable of expressing all the synchronization between rt-objects that an author may require. For basic temporal synchronization this does not represent any particular problem. We shall look more closely at the more difficult question of conditional synchronization. There are essentially 5 different cases that an author may call for:

6.4.1 Primitive Events

These are just instants or points of the T (time) axis. Examples are

1. During the presentation of a run-time rt-object, the presentation crosses a predefined time threshold, known in the MHEG standard as a "timestone", and this may cause a trigger to fire.

2. An action such as *set-volume* may trigger a link set to fire if the volume reaches a certain level.

3. A user makes a selection by clicking on a button or selecting in a menu.

Primitive events such as these may be specified directly using MHEG.

6.4.2 Disjunction ($\vee$) of Two Events

A disjunction of two events $E_1 \vee E_2$, is an event which takes place when either E_1 or E_2 takes place. This can be specified directly in an MHEG object, since the OR operator is available to combine the trigger conditions that cause a link to fire.

6.4.3 Conjunction ($\wedge$) of Two Events

A conjunction of two events $E_1 \wedge E_2$, is an event which takes place when E_1 and E_2 have occurred whatever the order. A conjunction event cannot be handled directly in an MHEG link since it would require the MHEG engine to memorize events, and this is counter to the "final form, minimal resource" philosophy of the standard. However the author need not despair, the following MHEG technique converts the conjunction into a disjunction, which an MHEG engine can handle easily.

The following objects are required:

1. C_1 and C_2 are two content objects each containing a generic value[4]

2. L_1 is a link which fires when event E_1 occurs. Its effect is to set the value "E1" into C_1.

3. L_2 is a link which fires when event E_2 occurs. Its effect is to set the value "E2" into C_2. Surprised?

4. Link L_3 fires whenever event E_1 or event E_2 occurs. However it has the additional condition that C_1 contain value "E1" and C_2 contain value "E2". The associated action is the required action for the conjunction of the two events.

[4]The MHEG standard does not provide for variables, however content objects containing what the MHEG standard describes as *generic values* may often be used to achieve similar effects. The difference between a generic value and a variable is that Part 1 of the MHEG standard does not provide operations such as $+$, $-$, $\times$ and $\div$.

6.4.4 Sequence of Two Events

A author may specify a sequence of two events by defining a link L_1 which in turn creates (*prepares* in MHEG jargon) another link L_2 when event E_1 occurs. Link L_2 is fired when event E_2 occurs.

6.4.5 Negative Events

A negative event is specified with respect to a specified time interval, i.e. the event did not occur during that time. In the technique described here, the interval is between some arbitrary moment and a later moment when link L_1 becomes *not ready*. This interval will be called the "life" of L_1. It is assumed that L_1 is *ready* at the beginning of the interval.

To express a negative event, the following objects are required:

1. C_1 is a content object containing a generic value.
2. L_1 is a link which triggers when event E occurs; the associated action is to set a value "E" into C_1, and then to destroy the link L_1.
3. Link L_2 triggers when L_1 becomes *not ready* (i.e. has been destroyed for any reason). However L_2 has the constraint that C_1 should **not** contain the value "E". This constraint ensures that the event did not occur during the life of L_1.

7 UMass Lowell MHEG Engine[3]

The section describes the implementation work performed by the authors in an effort to validate and evaluate the MHEG model[5]. This design is based on the committee draft of the MHEG standard.

The processing element that delivers MHEG objects to an application for presentation is referred to in the standard as an MHEG engine. Although the minimum functionality of an engine is defined in the standard[6], the design of the engine is left to the system developer. Informative text in the standard suggests the basic functions of an engine include object decoding and encoding and an interface with an application and presentation system for event handling and presentation actions. Our model is shown in Figure 4:

Figure 4: shows that the application controls the MHEG engine through control functions such as *play* and *pause*. In general an MHEG presentation is not a linear time line; consequently a *play* operation depends upon the setting of the current position. Functions for specifying the reference object, typically a composite object, as the current position are a necessary part of the API between the application and the engine. Additionally, the application is typically interested in state changes that result from presentation of objects and user interaction. These state changes can be obtained by having the application register callback functions with the engine.

The engine interfaces with the presentation services on the platform. The engine will request that an object be presented when the object's state is made ready. User interaction results in events from the presentation system to the engine which cause a state change in the interaction object that the user invoked. This state change is important to the scheduling function in the engine.

The engine manages retrieval and instantiation of objects in the engine. There is a default object which is loaded under application control. This typically leads to other objects to be loaded. As mentioned in [4], MHEG engine object retrieval is on-demand. Although not shown in the diagram, it is expected that an engine will in general not retrieve the content associated with media objects. Rather, it will depend on the multimedia system services layer of the platform to actually transfer video and audio data from the continuous media file system to the presentation server for those data types. An example of how such connections can be set up is given in [7]. Consequently, the MHEG engine is primarily concerned with the control objects, that is, the objects that define the composition and the interaction of the presentation.

The control of the engine is embodied in the scheduler which determines when objects are retrieved and presented. The scheduler depends upon state changes of currently active objects and the link objects in order to drive the presentation.

8 Summary

There has been a considerable research effort into modelling realistic hypermedia synchronization and real-time delivery. The MHEG working group believe it is important that the standard should show that an author is able to handle all the theoretical constructs.

[5] The MHEG standard is also being evaluated by development work in Europe and Asia.

[6] The standard provides the following conformance statement regarding MHEG engines:

> A conforming MHEG engine is one which interchanges conforming MHEG objects [sic] instances

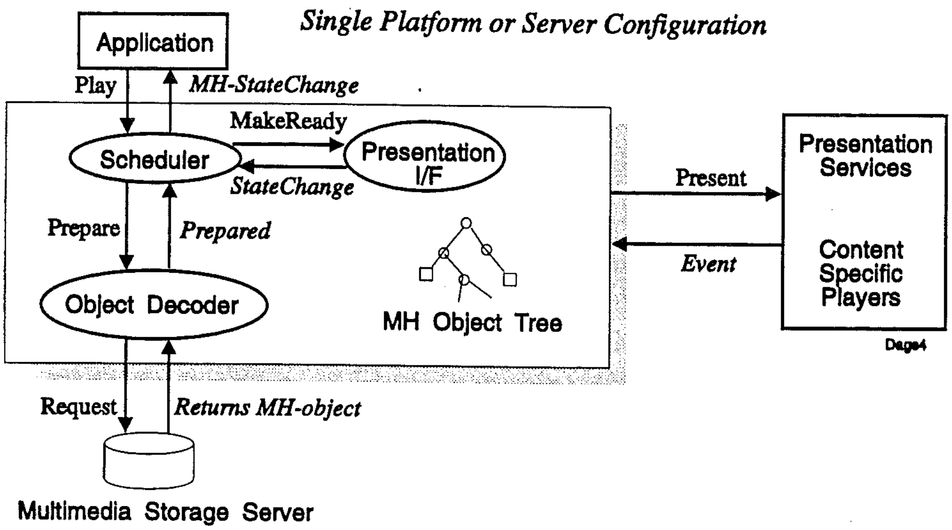


Figure 4: Umass Lowell MHEG Engine.

Clearly, in a practical situation, the hypermedia theory will be hidden from the audio-visual specialist, and will be the responsibility of the designer of an MHEG authoring system. We feel however that it is important that anyone reviewing MHEG should be aware of this work.

Many of the international experts who are developing the MHEG standard believe that the often announced, but yet to arrive, explosion of multimedia applications for the general public, can only occur when it becomes possible to exchange and re-use interactive multimedia and hypermedia information objects across heterogenous equipment and between different applications in real-time. Experience with RAVI and other wide area experimental architectures has shown that interactive audio-visual programming is a major investment, and the massive international effort required will only occur when investors are assured of the lasting effects of their investment.

The MHEG group believe that a strong consensus standard for multimedia and hypermedia objects is a vital pre-requisite.

9 References

[1] ISO/IEC. Draft International Standard 13522-1(MHEG): Coding of Multimedia and Hypermedia Information, Preparation document version v0.3, Sept. 12, 1994.

[2] Herzner, W. ISO/IEC. Draft International Standard 13522-1(MHEG) MCR 94/726: Text for describing uses of object-oriented concepts within MHEG.

[3] Buford, J., Gopal, C. Standardizing a Multimedia Interchange Format: A Comparison of OMFI and MHEG, Proc. International Conference on Multimedia Computing and Systems. May 1994, Boston, MA.

[4] Price, R. MHEG: An Introduction to the Future International Standard for Hypermedia and Multimedia Object Interchange, Proc. ACM Multimedia 93, Aug. 1993. pp. 121-128.

[5] ISO/IEC. Committee Draft 13522-1(MHEG): Coded Representation of Multimedia and Hypermedia Information Objects, Part 1, June 15, 1993.

[6] Koegel Buford, J. On the Design of Multimedia Interchange Formats, Proc. Third international workshop on Network and Operating System Support for Audio and Video, Nov. 1992.

[7] Interactive Multimedia Association. Multimedia System Services Version 1.0 — A Joint Submission from HP, IBM, and SunSoft. June 1993.

[8] ISO/IEC. IS 8824 Specification of Abstract Syntax Notation One (ASN.1). Second edition. 1990.

[9] ISO/IEC. IS 8825 Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1). Second Edition. 1990.ISO/IEC.

[10] ISO/IEC. IS 10744 Hypermedia/Time-Based Structuring Language (HyTime), Aug. 1992.

[11] The MHEG FAQ.

Legal Aspects of Electronic Publishing: Look Both Ways Before Crossing This Street

Glen M. Secor, Esq.

INTRODUCTION

Protection of intellectual property rights begins not when a work is published and placed into the market, but rather when the work is being developed. This paper will address some of the critical legal issues facing publishers and others in the acquisition and development of content for electronic publishing, including multimedia publishing. Special attention will be paid to the interests of publishers and authors in the various transactions involved in developing digital works.

Beginning with traditional book contracts and continuing through electronic publishing development agreements and multimedia joint ventures, the author will examine the emerging rights issues in electronic publishing. The focus throughout is on developing the business relationships and securing the rights needed to publish electronic works.

I. ELECTRONIC RIGHTS IN AUTHOR-PUBLISHER BOOK CONTRACTS

One approach to the acquisition of electronic publishing rights is to simply include them with the transfer of traditional print rights. Publishers have long sought to do this with the "all media now in existence or hereinafter discovered" clause of the typical book contract, meaning that the publisher acquired the rights to the book in print and electronic form. Ten or twenty years ago that clause may not have meant much to authors, or perhaps even to publishers. But with new media being developed on a near constant basis, and with electronic publishing seeming to be the wave of the future, electronic rights are no longer an afterthought in book contracts. Now publishers, authors, and agents are finding that electronic rights often do not fit neatly into the traditional book contract.

The National Writers Union has developed a "Statement of Principles on Contracts Between Writers and Electronic Book Publishers" (National Writers Union, 1993). This Statement is useful not only because of the specific positions being advanced by the NWU, some of which will be discussed here, but moreover for the list of issues which it addresses. These issues, using the NWU's section headings, are: 1. Copyright, 2. Grant of Rights, 3. Creative Control, 4. Manuscript Acceptance, 5. Royalties, 6. Royalty Statements, 7. Termination, 8. Option, 9. Non-competition, 10. Arbitration, and 11. Affordability and Access. This section will focus on certain of these topics, but the analysis is not limited to the issues specifically raised in the NWU proposal.

A. Copyright

Copyright is not an issue, per se, simply because a work is to be adapted to electronic form or because it is prepared originally in electronic form. Authors and other creators own the copyrights in the works which they create. They transfer the rights in their works, usually in return for remuneration, to publishers, movie studios, television studios, and others who are positioned to exploit those works. Electronic publishers, some of whom will also be print publishers, will be among the potential transferees of rights.

Copyright becomes more complex not because of the electronic publication of a work, but because of the potential for collaboration among creators and integration of various works which exists in the electronic environment. This is the whole essence of the "multimedia" movement. For electronic publishers, the new trick to copyright, if there is one, is in keeping track of who owns which rights in what elements of the electronic work. This task is obviously at its most complex in a true multimedia work combining various forms of content from a multitude of sources, but also must be managed for a pure text work for which there are multiple contributors or to which is added proprietary search software.

The NWU position on copyright in electronic publishing agreements is that the author should control copyright, as he does in the print environment, until he or she makes a complete or partial transfer of such

rights. The NWU proposal acknowledges "work-for-hire" situations as exceptions to this norm. As will be discussed in the sections below, work-for-hire and other types of author-publisher arrangements may become more prevalent in the electronic world.

Moral Rights

The issue of "moral rights" is looming larger in the U.S. copyright picture and poses particular problems in the electronic environment. Moral rights are essentially authors' rights in the paternity and integrity of their works. Moral rights are given more weight in other copyright regimes, particularly those of European countries, but the international nature of trade in intellectual property and the U.S. accession to the Berne Convention have increased their importance here. While U.S. law generally does not provide for explicit moral rights, the Visual Artists Rights Act, 17 U.S.C. 106A, does provide such rights for works of fine art. (Greguras. et al., 1994)

There is an undeniable tension between the legal trend toward moral rights and the practical reality of new information technologies. Because of the ease with which electronic information can be manipulated and passed along, it would seem more difficult for publishers to safeguard the moral rights of authors in the electronic world than in the print environment. Anyone who participates in online discussion groups or newsgroups has doubtlessly witnessed instances of manipulation or improper attribution of quotes.

As will be discussed below, in the section on clearing rights, publishers must be careful how they use copyrighted works in their electronic publications. What the publisher considers to be necessary and appropriate editing, perhaps by using only a portion of a film clip, still photo, or sound clip, might be unacceptable to the author or performer. This could be a problem even if licensor of the material to the publisher, say the movie studio or the art house, regards the use as acceptable. Remember that moral rights are author rights and do not travel with the copyright. The examples presented here might not pose a legal problem for the publisher in the U.S., but could elsewhere in the world.

The issue of moral rights must also be considered in the distribution of electronic works. Publishers need to be aware of their obligations to protect moral rights and must take reasonable steps to meet those obligations. What constitutes reasonable protection remains to be determined, but the possibilities range from the relatively simple to the extremely burdensome. When the publisher licenses a work for end use, or sublicenses it for inclusion in another publisher's work, it might be sufficient to include a statement of moral rights in the license, thereby placing the burden of protecting those rights on the licensee. When a publisher distributes material online, it might be necessary to encrypt the material to ensure authenticity. These matters need to be resolved as the law and the technology continue to develop in the coming years.

B. Granting and Termination of Rights

The granting and termination of electronic rights is clearly one of the biggest sources of potential conflict between publishers and authors in book contracts. As indicated above, the "all media now in existence or hereinafter discovered" clause does not fly anymore. Few authors are willing to give up such sweeping rights, especially to print publishers whose ability to produce and market electronic works is unproven at best. The difference between print rights and electronic rights is profound in this area.

When we talk about print rights to a book, even a trade book with significant commercial appeal, we are really talking variations on a single theme. Paperback rights, mass market rights, foreign rights, serial rights, reprint rights, etc., all involve the same fundamental product - a print book. The formats, for the most part, are long-established and do not change much. Movie rights and other dramatic rights in the story are generally handled separate of the print rights.

"Electronic print rights," though, are different. Electronic formats are evolving and will continue to do so. Publishers, whose ability to exploit the electronic formats of today is largely unproven, may or may not keep pace with the developments in technology. The publisher who develops a successful CD-ROM publishing program today may miss the boat on online publishing tomorrow. It is quite understandable that

authors are reluctant to enter into long-term transfers of electronic rights in the midst of such uncertainty.

Electronic rights also differ from print in that one of the great terminating events of the print contract, "out of print" status, perhaps ceases to be in the electronic world. When a print book is "out of stock" and no further printings are planned, the book is clearly out of print. It is difficult to classify as in print or out of print a book which exists is digital form and is accessible electronically.

In this context, the NWU proposals on the granting and termination of electronic rights seem to make sense. First, electronic rights should terminate if they are not exploited within a reasonable and stated period of time by the publisher. A publisher who does not develop an electronic publishing program, or who does not include a particular work within such a program, should not be allowed to sit on the electronic rights to that work (see also Curtis, 1991). Further, because publishers may not keep pace with the changes in technology, electronic rights should be granted for much shorter periods of time than the duration of the print rights. Finally, "out of print" status should be replaced by "out of promotion" status, meaning that the publisher is no longer marketing the work and rights should revert to the author.

The provisions, of course, may not seem quite so sensible to the publisher. In the electronic arena, the publisher is dealing essentially with the same uncertainties as the author. No one really knows how the technology and markets will develop. The economics of electronic publishing are very dicey, with many pointing to big potential profits down the road, but few (if any) earning them today. Publishers run huge risks of investing too much or too little in their electronic publishing programs.

Publishers also run risks in not acquiring electronic rights or in acquiring those rights for short durations. Take for example a professional book, science book, or textbook with significant backlist or revised edition potential. Assume that the title is expected to sell for ten years or longer. Assume further that no significant market exists for the title in electronic form today. Can anyone look even a decade into the future and be certain that no electronic market will develop for the title during that time? Probably not. But what if, because of all the concerns outlined above, the author is willing to grant the electronic rights for only five years? The publishing house could find itself in a position of losing the electronic rights around the time that the electronic market for the title develops. The publisher, after largely making the market for the title through sales of the print edition during the first five years of publication, could see the print edition facing competition down the road from someone else's electronic edition. And the publisher could lose the electronic rights even if it was reasonably positioned to exploit them, because of, say, an unrelated dispute with the author, or because the author honestly believes another company is better suited to bring out the electronic edition.

No, the segregation of print and electronic rights is not as simple or "fair" as it might first appear. And if we project far enough into the future, when a book might be available in any number of different formats and via many means (e.g. printed and bound, printed on demand, online, on CD-ROM or other storage media, etc.), it is hard to imagine a publisher making the primary editorial and marketing commitment to the book without having all or most of these rights during the return on investment period.

So, we find ourselves in an age of uncertainty over who should control the electronic rights to texts. Uncertainty breeds risk, and the essence of this dilemma is allocating the risks between publishers and authors. Various author and publisher groups have argued that one side or the other should always control electronic print rights, but it is doubtful that any such absolute approach can succeed. These matters, at least for the time being, seem destined to be negotiated on a case by case basis. The nature and electronic publishing potential of the work, the potential market for the work, and the expertise and track record of the publisher are some of the factors which must be taken into consideration in deciding who should get which electronic rights and for how long.

Sublicenses and Transfers

Of course, a publishing house, even if it holds the electronic publishing rights to various works, may not develop the capability to effectively exploit all of those rights. The markets for electronic books are just developing and few, if any, publishers will be able to establish themselves in all potential markets in the foreseeable future. Therefore, we will likely see a healthy market for sublicensing and transfer of these rights. Such a trend, though, could fly in the face of author desires, as indicated in the NWU proposal, to keep a tight leash on these rights and creative control over the electronic projects in which their works are used. Sublicensing or transfer by the primary publisher of the rights of all or part of a work may represent the author's best chance at having the work successfully exploited in the electronic marketplace. Still, authors may be reluctant to give publishers the unfettered right to enter into such sublicenses and transfers.

One approach to this problem would be to make transfers and sublicenses of the electronic rights subject to the approval of the author, which could not be unreasonably withheld. Some print book contracts contain such a provision regarding the primary publishing rights. Publishers, however, might find this requirement to be burdensome, especially if the volume of permissions and sublicensing is great. In that event, publisher rights and permissions departments will probably be struggling to keep up with the volume of requests, never mind having to clear each transaction with the author.

The International Publishers Copyright Council, in the report from its Third International Copyright Symposium (held in May 1994, in Turin), suggests a hierarchy of electronic rights which authors and publishers should agree upon. This hierarchy would consist of "prime rights" and "subsidiary rights" (International Publishers Copyright Council, 1994). Prime rights would be the right "to issue a copyright work on electronic media" (i.e. the right of the publisher to publish in electronic form) and the right "to authorise the storage of a copyright work in any medium by electronic means" (e.g. to allow a document delivery service to store a digital copy of the work for printing and delivery to its customers). Subsidiary rights would include the right of the publisher to include the work in "another publisher's/producer's electronic product or service" or in "multimedia works, and to authorize the downloading, distribution, or networking of the work by third parties, as well as certain other rights.

Publishers and authors could utilize this hierarchy of rights to determine which rights can be sublicensed or transferred by the publisher with or without the author's permission. The prime right "to issue a copyright work on electronic media," for instance, might never be transferable without the author's approval. The right of the publisher to license the work or a portion thereof for inclusion in another publisher's electronic product or in a multimedia work might require the author's assent in some instances but not in others. By recognizing the various prime and subsidiary rights, and by negotiating up-front any limitations on the publisher's ability to sublicense or transfer those rights, publishers and authors will avoid unnecessary surprises or disputes over the use of the published work.

C. Royalties

Of course, if the economics can be made to work to everyone's advantage, then concerns over the granting of electronic rights would diminish. Unfortunately, that situation does not exist today. If the early positions on royalties are any indication, economics may be the most contentious aspect of electronic publishing.

The NWU position on royalties is predictable and somewhat understandable: royalty rates on electronic books should be higher than on print books, to reflect the lower production costs on electronic books. Royalty rates on books sold online should be even higher, to reflect the lower costs of network distribution. After all, it is clearly cheaper to produce a copy of a CD-ROM disk than to produce a copy of a print book, or to transmit a book electronically versus shipping a print book.

The problem with this position is that it takes such a narrow view of production and distribution costs. Even if we assume for the moment that editorial and marketing costs are similar for electronic books and print books, and that the actual physical production and distribution of copies is cheaper in the electronic world, it still does not follow that electronic books cost less to publish than print books. There are development costs

associated with electronic books, including software and other technical development costs, which may not be present in print books. Even in the case of online distribution, there are tremendous costs associated with data storage and transmission. These costs cannot be overlooked in the demand for higher author royalties.

The NWU also advocates the payment of royalties based upon the list price of the electronic book, rather than on the net price, as is the case for print books. One exception to royalties-on-list would be for the sale of copies which are bundled by hardware manufactures for sale with the machines themselves. The rationale given for royalties-on-list is that royalties-on-net allows too much potential for publisher abuse and creates suspicions in the minds of authors.

There is a reason for the current royalties-on-net approach, of course. The book distribution and selling process in this country has utilized discount-off-list pricing, including now at the retail level. The list price persists, despite the fact that no one in the buying chain, except sometimes the individual consumer, pays it anymore.

If books, like most other consumer goods, came without a "manufacturer's list price," we would not be having this debate over the appropriate base for author royalties. But semantics aside, it is hard to understand why author royalties would be based on anything other than publisher revenues from the sale of the books. Publishers, one assumes, are in the business of maximizing their revenues per title. Discounts are given, presumably, increase the volume of books in the distribution chain and to maximize dollar sales. Author royalties are maximized when publisher sales are maximized. Publisher sales, like sales for every other manufacturer, distributor, retailer, or service company, are booked at net.

As unsatisfying as the NWU position on royalties on electronic books might be, it is no more so than that reportedly adopted by Harcourt Brace for the electronic versions of some of its print books. A dispute between Harcourt and one of its authors, NWU member Larry Bailey, has resulted in a lawsuit. Harcourt's is asserting that it is not obligated to pay royalties on the digital versions of two accounting books which Bailey wrote or cowrote, as Bailey's contract specifies royalties only for print editions (Reid, 1994). Harcourt took this position after Bailey had rejected Harcourt's offer for a substantially reduced royalty for the digital edition.

What is most interesting about this dispute is Harcourt's rationale for pushing the reduced royalty in the first place. According to a Harcourt spokesperson, the reduced royalty is needed because "electronic versions have significant development costs and the software firms that designed them are assigned a royalty as payment." (Reid, 1994)

Yes, but what about the lower production and distribution costs which the NWU cites as justification for higher author royalties on electronic books?

Electronic publishing is a relatively new game and its economics are unclear. One cannot help but sense a bit of opportunism by publishers and authors in the face of this lack of clarity. Each side is pointing to the elements of the cost equation which support its argument for higher or lower author royalties. Neither side seems to be acknowledging the entire cost equation, however uncertain it might be.

What is needed here, rather than opportunism, is realism and an open sharing of information. The electronic rights clauses of many, many existing book contracts are ticking time bombs. The importance of electronic rights and royalties in book contract negotiations will only increase in the future. If either side goes too far in trying to exploit the situation, in individual transactions or in the aggregate, author-publisher relations will suffer immeasurably. The evidence thus far indicates that both sides may be headed in that unfortunate direction.

II. THE ART OF THE MULTIMEDIA DEAL

A. Work-for-hire and Development Agreements

Electronic publishing forces us to think not only about the terms of the traditional book contract, but also about the very nature of the author-publisher relationship and the role of each in the process. In the print world, most author-publisher transactions involve the arms-length transfer of rights and money, although a different relationship, usually under a work-for-hire arrangement,

is possible. In the electronic environment, publishers will find themselves entering into many more transactions which do not involve the outright acquisition of intellectual property. Some of these will consist of the licensing of content for specific and limited purposes, which is a presentation unto itself and will not be dealt with here.

Other transactions will actually increase the publisher's control over the content of its digital publications. These arrangements, essentially a variation of work-for-hire, are accomplished via development agreements. Print publishers who want to become successful electronic publishers will have to become familiar with the use of development agreements.

The role of the author into the digital and multimedia world is the source of much speculation and debate within the publishing industry. Some experts have predicted that authorship will change from a process of linear story telling (or explanation of facts) to one of multimedia integration, with the "text" serving primarily to navigate the "reader" through new multimedia worlds (Curtis, 1991). Under this view, the author of text will function not as an independent creator of content, but as part of a development and production team, with others contributing graphics, sound, interactivity, etc.. The product produced by this team could be influenced equally by any of the team members; the "story" would not necessarily control. One could argue that such an approach to authorship already exists in movies and television, with very spotty results.

This change in the nature of authorship may or may not occur on a broad scale, but it is already happening in subtle ways even in today's relatively simple digital text publishing projects. For example, in the Harcourt Brace-Bailey dispute mentioned above, the author is apparently doing annual updates on the books under a work-for-hire contract. As also mentioned above, Harcourt is pushing for lower author royalties because it must also pay royalties to the software firms involved in the project.

This type of arrangement, which is essentially an ongoing development and production team for these electronic books, will become more common as the volume of digital text publishing grows. The media in this case is the floppy disk. CD-ROM versions of the book would likely bring more and different players to the team, as would online publication.

In the context of author-publisher agreements, we must keep in mind that electronic publishing involves more than simply "printing" the text in another format. Digitization of a text does not alter its fundamental characteristics, but adding hypertext links does. Adding graphics and sound changes the nature of the work even further. When the objective of a project is not to simply digitize an existing textbook, for instance, but rather to develop an interactive CD-ROM for the teaching of a subject, one dimension of which might be the material contained in that textbook, then the role of the text and its author have changed fundamentally.

Development agreements are one means by which a publisher can specify and coordinate the roles of the various parties involved in an electronic publishing project. They can be used to control the work of in-house personnel, under work-for-hire arrangements, as well as of independent suppliers. Software publishers have long used them for software development projects, which tend to involve both in-house development, outsourcing, and third-party licensing. Electronic publishing projects in this sense can be managed similarly to software development projects.

An in-depth discussion of development agreements and the many provisions which can be included in them is beyond the scope of this paper. For a solid explanation of such agreements, including a good sample agreement, I recommend the handbook *Multimedia: Law and Practice* by Michael D. Scott (Prentice-Hall Law and Business, 1993). In fact, I will use the format of Mr. Scott's analysis in the following overview.

Successful electronic publishing development agreements begin with effective functional and detailed specifications of the product. The specifications, which can be used internally and with any outside party participating in the project, must clearly state not only what the product is to do, what it is to look like, and how it is to work, but also how the development project itself is to be accomplished.

Product specifications might include a general description of the title, the media and operating

system(s) on which the title will run, the number and types of graphics expected to be incorporated, the expected search and linking capabilities, the volume of text to be included, expected printing and downloading capabilities, rough screen layouts, compatibility with word processors and other types of software, networkability, packaging, after-sale customer support, et al.. Project specifications would address such issues as file formats, security, documentation, testing, training, deadlines, budgets, confidentiality (i.e. re: product information and trade secrets learned during development), change procedures, etc..

Many of these specifications would be incorporated, directly or via addendum, in the development agreement. Their utility is less legal than practical, though. Whatever the nature of the electronic publishing project, whether it is a simple text on floppy disk or a full-blown multimedia CD-ROM, and no matter what combination of in-house and outside resources are being used, the specifications serve as the map for the project, indicating both the destination and the route which is to be taken. It is, of course, essential that any changes made to the specifications over the course of the project be communicated to all participants promptly.

The development contract or contracts must clearly delineate responsibility for all aspects of the project and for ongoing maintenance of the product. Who is responsible for writing the captions for any still photos which are used in the title? Who is responsible for updating the text and graphics? Who is responsible for customer technical support? Who is responsible for making obtaining all necessary licenses and clearances for copyrighted material being used in the work? What happens if the search software ceases to run or run effectively on future generations of operating systems? Even in the simplest of electronic publishing projects, the publisher, author, and software supplier(s) must know who is responsible for what.

Beyond specifying the product and allocating the various project responsibilities, the biggest issue facing the parties to an electronic book development agreement is sorting out who will own what aspects of the final product. Ownership of any content which is licensed for use in the product will be clear and will be governed by the licenses. Ownership of content or functional software which is developed specifically for the product may be less clear. Does the author of the text own the text which she writes for inclusion in the electronic book, or was she brought in on a work-for-hire basis, with all rights to the text being owned by the publisher? If the text includes forms, say accounting forms, who owns the software which the outside software firm develops to allow end users to fill out forms online?

There are legal tests and standards which a court can apply to the specific facts and circumstances of a case to settle disputes over ownership. The parties to the development agreement should avoid such disputes altogether by agreeing from the outset who will own the various components of the final product.

Ownership issues are critical to pricing, as well as use. If the software firm is to own the forms software at the end of the project, and if it believes that it can sell that software to other publishers for use in their electronic accounting texts, then the firm is likely to charge less for developing the program than if it does not own the software. If publisher does not want to have other publishers using the program, it will presumably be willing to pay more in development fees in order to secure ownership. But if the software firm is to own the software at the end of the project, the publisher should insist on some sort of license which allows it to use the program for a certain period of time without having to pay any additional licensing fee.

The parties are free to allocate ownership in the elements of the final product any way they choose. They should do so right from the outset in order to avoid misunderstandings and disputes.

My point in this section has not been to identify all of the issues which must or should be addressed in an electronic publishing development agreement. Rather, the intention has been to indicate the complexity of multi-party development projects, and to show how they differ from a simple two-party author-publisher book contract. On some electronic publishing projects, all transactions will occur at arms length and issues of responsibility and ownership will be clear, i.e. when the publisher acquires the content from the author, licenses the software from the software firm, then handles the tasks of digitizing the text and integrating the software in-house. There, the legal agreements between the parties can be very straightforward. In other

circumstances, as I have tried to indicate, the transactions, along with issues of responsibility and ownership, will be more complicated. The best response to this complexity is a clear and comprehensive development agreement.

B. Acquiring and Clearing Electronic Rights

Electronic publishing is essentially the marrying of content and software. Multimedia publishing expands the types of content significantly and the sources for that content exponentially. Because of the software component and the addition of non-text media to the content mix, intellectual property issues in electronic publishing are vastly more numerous and complex than in traditional print publishing. A full review and legal analysis of these issues is beyond the scope of this paper, but publishers should be aware of the following major points.

As has been discussed above, electronic publishing projects are likely to involve collaboration with software firms and developers, as well as with other providers of content. Some electronic publishing programs, as well as individual electronic publishing projects in other programs, will be structured as or at least will function similarly to joint ventures. Each party to the venture must realize the following: the rights of the venture in the content which it publishes will only be as good as the rights of the party which contributed that content. Further, if the work produced by the venture is found to infringe or violate the intellectual property rights of a third party, the venture and not merely the party which contributed the infringing material will be liable. This means that each participant in an electronic publishing venture must be confident not only in its own rights in the property which it contributes, but also in the rights of all of the participants in the property which each contributes.

When business enterprises agree to form a joint venture, the parties to the joint venture generally conduct some form of due diligence on each other. This might involve looking into the finances of the other joint venture partners, interviewing the various management teams involved, physically inspecting one another's assets, etc.. When companies and individuals come together in an electronic publishing venture, whether it be an ongoing publishing program or just a single publishing project, they should be concerned about the financial health and other key characteristics of the partners, but should also conduct due diligence on the intellectual property being brought in to the venture.

In an excellent article entitled "Intellectual Property Due Diligence for Multimedia Strategic Alliances," William Tanenbaum explains not only why such due diligence is necessary, but also what it should seek to accomplish (Tanenbaum, 1994). Atty. Tanenbaum's 12-point checklist of subjects to be addressed in due diligence is so valuable that it is duplicated here:

> 1. Whether the venture party owns the contributed property, or has a license right sufficient to grant the alliance the right to exploit the property in the intended manner.
>
> 2. Whether the intellectual property rights in the contributed property are valid and enforceable.
>
> 3. Whether any third party has any intellectual property rights in the property, and if so what the nature of the interest is.
>
> 4. Whether the contributing party is bound by any agreement, obligation or restriction which would prevent it from granting the intended rights in the property contributed to the venture.
>
> 5. Whether the contributing party (or other owner) has "perfected" the intellectual property rights through proper registrations, recordations, and other filings in the United States patent, copyright, and trademark offices, and if applicable, in offices of foreign governments.
>
> 6. Whether there are defects in any such filings which need to be corrected.
>
> 7. Whether intellectual property rights have been and are being properly kept in force through the timely payment of patent maintenance fees and the like, both in the United States and abroad.
>
> 8. Whether the venture's exploitation of the contributed property will infringe an intellectual property right of a third party.
>
> 9. Whether the contributed property is the subject of any past, pending, or threatened litigation, and if so, what the effect of this is, or is likely to be, on the venture's intended marketing of the contributed property.
>
> 10. Whether there are any aspects of the contributing party's past or current licensing practices which give rise to patent or copyright misuse or which would otherwise render the

intellectual property rights in the contributed property unenforceable.

11. Whether the contributing party has obtained proper United States and foreign moral rights waivers, permissions to use actors' likenesses, and permissions from entertainment industry guilds, unions, and the like.

12. Whether the contributed property is subject to any existing or contingent security interests or similar encumbrances.

I strongly encourage the legal or rights department of every publisher to pick up a copy of the complete text of this article and to consider a subscription to The Computer Lawyer, which has included a number of excellent articles about multimedia rights in recent months.

Notice the breadth of intellectual property rights mentioned in the checklist, as well as the various aspects of these rights which should be investigated. The introduction of software to the equation raises the issue of patents, something with which most print publishers have probably not dealt. Inclusion of stills, film clips, and sound clips necessitates consideration of right to likeness, right of publicity, trademark, etc.. Intellectual property protection in electronic publishing encompasses more than traditional copyright protection.

III. FACILITATING THE LICENSING, SUBLICENSING, AND END-USER LICENSING OF ELECTRONIC WORKS

Time and space do not allow for fuller consideration of this topic, but the following sources should be of interest to publishers and others involved in electronic publishing.

The Copyright Clearance Center (CCC) "Rightsholder Electronic Access Agreement," which establishes CCC as a possible clearing house for electronic subrights pursuant to the appropriate grant of rights to CCC from the publisher.

Mark L. Gordon and Timothy P. Walsh, "Transaction-Based Licenses: Managing Revenues and Controlling Costs," The Computer Lawyer, v. 11, no. 10 (Oct. 94).

Fred Greguras and Sandy J. Wong, "Software Licensing Complements the Digital Age," a paper made available by the Electronic Frontier Foundation (EFF) via the World Wide Web (`http://www.eff.org`) or by contacting EFF and requesting a copy.

"Negotiating Networked Information Contracts and Licenses," a draft paper prepared by Robert Ubell Associates for the Coalition for Networked Information (CNI) as part of CNI's READI program (Rights for Electronic Access to and Delivery of Information). This outstanding document, dated Nov. 15, 1994, examines a comprehensive set of terms and issues which should be addressed in any networked use license. Available from CNI via the WWW (`http://www.cni.org`) or by contacting CNI and requesting a copy.

IV. SUMMARY

This paper has addressed a number of legal and business issues relating to the acquisition of intellectual property for the purposes of electronic publishing, at best scratching the surface of any of them. Electronic rights will be one of the most difficult issues publishers will face in the electronic world. A successful transition from the legal practices which have developed in the print world to those which will be required in the electronic world will depend upon the degree of openness and understanding which all involved, including authors, publishers, software firms, and others, bring to the table. Some of the issues involved with electronic publishing will resolve themselves easily, while others will require ongoing negotiation and careful balancing of a number of potentially competing interests. Awareness of the law and the underlying business realities will be essential to the development of a sensible legal framework for electronic publishing.

REFERENCES

Richard Curtis, "Here Come the Cyberbooks: Future of Publishing Glimpsed Through New Contract Clause," Locus, the Newspaper of the Science Fiction Field, 1991.

Fred Greguras, Michael R. Egger, and Sandy J. Wong, "Multimedia and the Superhighway: Rapid Acceleration

or Foot on the Brake?," The Computer Lawyer, v. 11, no. 9 (Sept. 94).

International Publishers Copyright Council, "The Publisher in the Electronic World" (a report for The Third IPA International Copyright Symposium), 1994.

Calvin Reid, "NWU Calls Harcourt Unfair in Digital Royalty Dispute," Publishers Weekly, Dec. 12, 1994.

William A. Tanenbaum, "Intellectual Due Diligence for Multimedia Strategic Alliances," The Computer Lawyer, v. 11, no. 10 (Oct. 94).

Transaction Protection for Information Buyers and Sellers

Steven Ketchpel
Robotics Laboratory
Stanford University
Stanford, CA 94305
ketchpel@cs.stanford.edu

Abstract

Although existing payment mechanisms protect the parties from snoopers who might be intercepting network messages, most do not provide much protection from misconduct by the other party involved in the negotiation. We present three different approaches that address this deficiency. The first relies on the message delivery level for automatic acknowledgment of messages. The second makes use of a trusted third party which acts as an intermediary for the transfer of information, so that the seller can prove that the information was sent (even if the buyer claims it was never subsequently received.) The third approach provides greater security, enabling the prevention of fraudulent transactions, rather than just providing proof after the fact. This approach places greater demands on the third party, essentially turning it into an escrow agent. Finally, we show how these approaches can be integrated with existing payment mechanisms.

1 Introduction

Historically, the Internet evolved from a U.S. government sponsored project to enable researchers to share information. In a research environment where progress depends on the flow of ideas, the culture discouraged barriers to the exchange of information, and researchers were typically pleased to share their results with others in their community at no charge. A relatively small number of commercial on-line services charged using a connect-time and pay-per-item structure. Since each user typically used only a handful of these services, it was reasonable for the users to have a separate account with each service, and pay bills by traditional means, such as mailing a check every month. In the last few years, however, the Internet has begun a transition from a research tool to a commercial environment. It is increasingly easy for an individual to become an information provider reaching a huge available audience.

In some cases, providers have been willing to continue in the paradigm of making materials available at no charge. For example, product literature is generally available without a fee. Until information providers can be compensated by the consumers of their products, however, the range of contributors will be limited to those who have commercial interest in distributing (e.g., marketers and politicians) or those with sufficient slack resources (both time and hardware). When the market is entirely driven by producers, there may be a mismatch between what is available and what consumers find useful.

The last year has seen a flurry of activity as companies try to be the first to provide a general charging mechanism requiring a minimum overhead difficulty and expense. All of them recognize that in the open environment of the Internet, there is the serious risk that nearly any message on the network can be intercepted and read by an eavesdropper. Therefore, these payment mechanisms afford some type of protection against malfeasance by a party outside the transaction. However, most payment mechanisms do not safeguard against bad faith behavior by one of the transaction participants.

This paper considers three different approaches to providing a greater level of security for the parties involved in the transaction. All three share the underlying mechanism of an explicit contract and the means to prove or disprove several modes by which either party may fail to keep its end of the contract. The first approach relies on support features at the level of message transport. Since these features are not existent in all platforms, we consider the second and third alternatives. The second works on a "catch and punish" model, employing a trusted third party, which acts as a reliable delivery service. The third places greater demands on the trusted third party, putting it in the role of an escrow agent. The next

section of the paper discusses the purpose of the contract and the types of fraud it is designed to prevent. The third, fourth, and fifth sections present the detailed protocols which achieve these ends in each of the three approaches. The sixth section shows how the new protocol may be integrated with existing payment mechanisms.

2 The Contract–Whom Does It Protect?

Like any traditional sale of physical goods today, the key point of an information sale is that the buyer gets the requested information and the seller gets the agreed-upon price. However, there are essential differences between information and physical goods. Most sales today have a warranty or exchange period during which the goods can be returned, with the buyer receiving a full refund of his purchase price. In the domain of information, this may provide inadequate protection for the merchant. The buyer's ability to make a perfect copy before "returning" the goods places a large risk on the merchant who allows full refund returns. Indeed, many stores do not allow the return of software products after they have been opened for this reason. A second difference is that the buyers don't take possession of the goods at the same time as they make payment, as occurs in most face-to-face sales. Phone and mail orders share this difficulty, but retailers have the option to send goods by registered mail, so the merchant has proof of the buyer's receipt of a package (although the contents are not verified by this mechanism).

Given the risk to the merchant of a customer retaining a copy of the information when applying for a refund, we advocate charging the customer when the information is received, and in general not permitting refunds in the case where the requested information was provided. In order to protect the customer, then, there needs to be a mechanism for the customer to prove that the information he received did not coincide with what the merchant promised to deliver. There also needs to be a forum in which such complaints can be resolved in a timely and cost-effective manner. The protocols presented in this paper provide a mechanism which allows customers to prove that received information did not match what the seller agreed to provide.

On the other hand, the merchant must be protected to ensure that the customer cannot benefit by denying receipt of the information. The protocols presented here give a way for the merchant to show that the requested information was sent. If the acknowledgement is built into the message delivery system, this also proves that the buyer received the information.

In some applications, the problem of denial of receipt is not serious. since the merchant can retransmit the requested information at relatively low cost. In other domains, the problem can be more serious. Time-dependent data such as stock quotes cannot always be simply re-sent. A further problem is the delivery of electronic cash, such as DigiCash [2]. DigiCash is explicitly designed so that the issuing bank is unable to associate a particular cash token with a user, using blind signatures. If a user denied receiving her cash, the bank would still have to honor the "missing" tokens, since it has no way to determine which ones were in the supposedly lost mail. The bank must make certain that the customer receives the electronic cash token, and the protocols described below could assist it.

What we really want is a protocol that satisfies the following specification:

1. Both the buyer and seller have contracts for the agreement to buy the specified information at a specified price, signed by the other party.
2. If the buyer paid the seller, then the buyer can prove that he paid the seller.
3. If the buyer received the information, then the seller can prove this.
4. The buyer can prove that the received goods were not the promised goods, if that is the case.

Unfortunately, it is not hard to show that this specification is typically unsatisfiable. In particular, the third condition cannot be satisfied if the seller's only proof that the buyer received the information is an acknowledgment from the buyer. The buyer can simply refuse to send the seller any further messages after getting the information. (Notice that although the second requirement seems very similar to the third, it is typically possible for the buyer to prove that the seller received payment without receiving messages from the seller. For example, the buyer may be able to point to a Visa statement.)

3 Approach #1: Verification at the Transport Layer

One option is to build an automatic unforgeable acknowledgment mechanism into the message-delivery

system, which provides proof that the message was received, obviating the need for action on the part of the buyer to inform the seller of the receipt of information. If the buyer denied having received the information, the seller could simply produce the acknowledgment that was returned to her when the transport mechanism delivered the message to the buyer. While this may be an interesting feature to build into the message-delivery system, we cannot count on its presence. Even if the seller can prove that the message was received, the contents of the message are not verified. The buyer can claim that the message which was received (and acknowledged) did not contain the requested information. Similarly, an unscrupulous merchant could send a bogus message, then, when challenged in court, claim that a different message (one which provided the requested information) was sent. Therefore, the transport layer would need to provide the unforgeable acknowledgment stating not only that a message was received, but also what its content was. Since we know of no transport mechanism that provides this capability, one possibility is to add this as a third party service.

4 Approach #2: Third Party "Registered Mail" Service

In this approach, messages make their way between the buyer and seller via a third party, e.g., a commercial service that is recognized for its impartiality and trustworthiness. The third party records not only that the message was sent, but also what its contents were, so that if a dispute should arise in the future, an impartial party can produce a record of the transaction. The fact that this service is not directly integrated with the message delivery level does result in a loss of certainty. Specifically, if we assume that the network is not perfectly reliable, a record showing that a message was sent from the seller, received by the third party and re-sent to the buyer does not prove that the buyer received the message (as an automatic, unforgeable receipt from the message delivery layer would). Therefore, we weaken our expectation from condition 3

3. If the buyer received the information, then the seller can prove this.

to the less demanding

3'. If the seller sent the information to the buyer, then he can prove this.

The heart of the protocol suggested here is quite simple: public key encryption is used for the buyer and seller to digitally sign a contract; standard electronic banking methods are used to transfer funds from buyer to seller; and the seller sends the encrypted information to the buyer via a third party that stores a copy (or digest) for future verification should it prove necessary. A more detailed description of the protocol follows, including a brief introduction to a key component, public-key cryptography.

4.1 Public-Key Cryptography

Public-key cryptography exploits the difficulty of inverting some mathematical function to ensure the security of the cipher. One of the most common schemes [4] is based on the difficulty of factoring large numbers. Encryption and decryption are accomplished through modular arithmetic. Each person has two keys that together satisfy certain mathematical properties. One key is called the "secret key" and is known only to the user. The second key is the "public key", and is published so that anyone can use it. Applying the public and secret keys are inverse operators, so applying the public key to the text yields an encrypted message that can only be recovered by applying the secret key, which should be held only by the intended recipient. If the keys are applied in the other order, with the sender encrypting a message with his own secret key, then any recipient can decrypt it by applying the sender's public key, but the recipient knows that only someone with the access to the sender's private key could have mailed the message. This option essentially acts as a "digital signature" offering some assurance that the message did originate from the supposed sender.

4.2 Protocol Step-by-Step Trace

1. The buyer obtains a description of the desired item and the sale price. This offer should be digitally signed by the merchant, as it forms the basis for the contractual agreement between buyer and seller.

2. The buyer digitally signs the contract, keeping a copy and sending a copy to the seller as his order. At the same time, the buyer initiates payment using one of the numerous payment options that are becoming available on the network. For instance, the buyer may provide a traditional credit card number, a charge account with one

of the billing agencies handling electronic transfers, or a digital cash token. This payment is encrypted with the seller's public key.

3. The seller processes the payment, forwarding any required information to the billing agency, and asks for approval of the transaction amount.

4. When the seller receives notification that the payment was honored,[1] he sends the information (encrypted with the buyer's public key) to the third party, along with the contract number. The third party archives the message (though since the information is encrypted, the third party is unaware of the true contents) along with identifiers of the involved parties and the identifying contract number. The third party then forwards the message to the buyer.

5. The buyer receives the information from the third party and decrypts it using his secret key.

However, this clearly places a large storage burden on the third party, who must archive the contents of all the messages sent to buyers. Fortunately, mathematical theory also provides a capability related to data compression that maps any message to a constant length called the message digest. Obviously, some information must be lost, so it is not possible to reconstruct the original message from the digest. However, it is possible to determine with near certainty whether a given message is the same as one earlier "digested." It is also extremely difficult given the digest to produce a different message which yields the same digest. In technical terms, the digest function is a one-way hash function. MD5 has been suggested as a digest function.[5] Using such a message digest feature, we can push the storage burden back to the selling party, while still retaining independent verification that the message produced is really the one that had been sent earlier.

4.3 Evaluating the Protocol

We now show that this protocol satisfies the specification (with condition 3 replaced by the weaker 3′). Both buyer and seller have copies of the contract signed by both parties, and by the properties of a digital signature, it must have been authorized on both sides by someone with access to the secret key. The buyer knows that the merchant received the money by the fact that his account has been debited. Since the payment was encrypted with the seller's public key, only someone with access to the seller's secret key could have decrypted it to deposit it. While no one is able to prove that the buyer received the information (because a network failure may have prevented delivery from the third party), the seller can prove that the information was sent. The seller merely needs to produce the message that, when encrypted and digested by the third party, matches the associated digest stored by the third party. If the buyer feels the received goods do not match what was contracted for, he can show the encrypted message (which the third party verifies is the one that the seller sent by comparing the message digests), decrypt it in court, and demonstrate that the contents do not fulfill the contract which was digitally signed by the seller. If the buyer acknowledges the receipt of the message, but claims that he could not decrypt it, the seller must show that, when the clear text message is encrypted with the buyer's public key, the resulting message matches the message digest stored by the third party.

[1] For a more detailed description of an architecture which modularizes the payment aspects of a transaction, see [1].

5 Approach # 3: Third Party as an Escrow Agent

While the previous approach provides the protections desired, they are all after the fact. In effect, a "catch and punish" strategy is used, rather than one of prevention. While there are means to tell which party did not live up to the contract, any resolution would need to take place after the buyer has obtained the information or after the seller has obtained the money. In an environment where connections are transitory and customers or even merchants can disappear quickly, after the fact protection might be insufficient. An alternative approach calls for the third party to play a more active role, validating the information and payment before sending the payment on to the seller or the information to the buyer. This method requires that the third party learn the contents of the messages and have a way to determine whether the information provided matches that which was contracted for. In the general case, this could require human intervention, but in specific cases it may be possible to do this authentication automatically. For example, if the transaction were a sale of shares of stock with an electronic transfer, it seems reasonable that the escrow agent could automatically check with the issuing agency, validate the certificate numbers, and even check that the certificate numbers are owned by the seller.

The escrow agent approach also allows the buyer to remain anonymous. Since the information is not

sent to the buyer until the payment has been verified by the escrow agent, the seller does not need to worry about receiving invalid payment. This method protects the buyer, too. Without the escrow agent, a buyer might be expected to tender payment before receiving the information. An unscrupulous seller might take payment but never send the information, secure in the knowledge that the anonymous party would have to reveal himself or herself in order to get restitution.

5.1 Protocol Step-by-Step Trace

1. The seller sends a digitally signed offer to the buyer.
2. The buyer signs the offer, retains a copy and returns a copy to the seller. The buyer also sends payment to the third party escrow agent.
3. When the seller receives the signed contract, she sends the requested information to the third party escrow agent.
4. When the escrow agent receives the messages from both parties, an evaluation is run to see that the terms of the contract are met by both sides. If so, the escrow agent forwards the information to the buyer and the payment to the seller. If not, the party not satisfying the contract is given a chance to re-send a satisfying message. Otherwise, the escrow agent notifies both parties that the contract has been nullified.

6 Integrating With Existing Payment Mechanisms

A number of systems for payment over a computer network have been proposed or are in place. In this section, we describe some of the representative systems and show how the transaction protection approaches described above could be applied. First Virtual [6] is an implemented system that has been operating for several months. The Simplified Network Payment Protocol [3] does not yet have a commercially operating implementation.

First Virtual (FV) acts as a clearinghouse, with each customer and merchant setting up an account—charges made against the account are collected by First Virtual, which levies a charge against the user's traditional credit card when a certain amount is reached. For each information transaction, the buyer sends his FV account ID to the seller in clear text. The seller can confirm that it is a valid account (though not the customer's ability to pay). The seller sends the information. FV inquires of the customer after he or she has the information whether the charge is to be approved. The customer may say yes, no (for any reason) or that the charge is not recognized and appears to be fraudulent. This system is a risk for the merchant, as a customer can refuse to pay for any purchase (though patterns of abuse will result in the account being closed).

By adding the digitally signed contract as a preliminary step (so that the purchase request is a contract) and sending information via a third party that archives a digest of the message contents, the merchant has recourse if the customer decides not to authorize payment. The merchant can prove that the requested goods were sent, and seek remuneration for them. Encrypting the account information when it is presented for payment also eliminates the hazard of "sniffers" that eavesdrop for account numbers on an insecure network. Since First Virtual is so accommodating to its customers, always giving them the right to refuse to pay for any purchase, the customers do not benefit from this added assurance.

The Simplified Network Payment Protocol (SNPP) requires that a customer place a "hold" against the funds in his account for the amount of the sale. The bank informs the merchant if the account has sufficient funds to cover the hold. If so, the merchant sends the information. After the customer receives the goods, he is expected to send an order authorizing the disbursement of the held funds. If the merchant doesn't receive her payment in a timely fashion, she can appeal for arbitration before the hold expires, ensuring that the customer does have the ability to pay. However, no provisions are made to prevent the customer from denying the receipt of the information. By enhancing the SNPP with the contract mechanism, both sides would have evidence to support their cases should the issue come to arbitration. The combination of the contract and evidence from the third party allows the claim that the information was sent to be verified, but also allows the buyer to demonstrate that the received goods were not the requested ones.

7 Conclusion

We have proposed three related mechanisms which may be added to payment protocols to allow a higher degree of confidence that an electronic information transaction will work out satisfactorily for both buyer and seller. In all cases, a digitally signed contract held by both parties prevents disagreements about

whether the information the buyer received is what he was expecting. The first approach of an enhanced message delivery system is both the simplest and provides the greatest assurance (that the buyer actually received the information). Until such powerful transport systems are universal, however, alternative methods are required. The second approach of a trusted third party acting as a "registered mail carrier" allows both buyer and seller to prove what was sent or received, even though the third party doesn't know the decrypted contents of the message. The third approach increases the expense to the third party, moving it into the role of an escrow agent. This approach does, however, provide the advantage that a transaction occurs only if the goods and payment have been first verified by the escrow agent. In cases where the information is of extremely high value or the parties have reason to mistrust each other, this approach may be the method of choice. Future work includes implementing these systems and integrating them with other digital library services.

7.1 Acknowledgments

This work was formulated as the result of discussions with the "Economic Issues" subgroup of the Stanford Digital Libraries, including James Kittock, Martin Röscheisen, Steve Cousins and Hector Garcia-Molina. Further discussions with Joseph Halpern were essential in shaping the final product. Thanks also to Robert Bosch, Steven Brenner, Cynthia Dwork, and D. Kusnan for their contributions. The author gratefully acknowledges support from the Office of Naval Research ASEE fellowship and NSF/ARPA/NASA under Stanford's Digital Libraries grant.

References

[1] Cousins, S., Ketchpel, S., Paepcke, A., Garcia-Molina, H., Hassan, S., and Röscheisen, M. InterPay: Managing Multiple Payment Mechanisms in Digital Libraries. Submitted to Digital Libraries '95.

[2] DigiCash. DigiCash brochure. 1994. Available at `http://www.digicash.com/publish/digibro.html`.

[3] Dukach, S. SNPP: A Simple Network Payment Protocol. MIT Laboratory for Computer Science Technical Report. Available at `ftp://ana.lcs.mit.edu/pub/snpp/snpp-paper.ps`

[4] Rivest, R.L., Shamir, A., and Adelman, L. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM 21*, 2 (February 1978), 120–126.

[5] Rivest, R.L. RFC 1321: The MD5 message digest algorithm. Internet Activities Board. April 1992.

[6] Stein, L.H., Stefferud, E.A., Borenstein, N.S., and Rose, M.T. The Green commerce model. First Virtual Holdings Incorporated Technical Report. October 1994. Available at `http://www.fv.com/tech/green-model.html`.

A Copyright Management System for Networked Interactive Multimedia*

John S. Erickson
Interactive Media Lab
Dartmouth Medical School
and
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755
oly@dartmouth.edu

ABSTRACT: This research will demonstrate how copyright permissions can be applied and extended in a secure, hierarchical fashion to various elements and composites in a network-deployed interactive multimedia production. Unique aspects of this work will include the broad application of the permissions extension concept to multimedia presentations, consisting of heterogeneous data objects, using a uniform document format; the development of a rights encapsulation kernel, **LicensIt™**, which will provide integrated support for hierarchical permissions extensions in the production environment; interactive, networked rights registration and clearance based on electronic licensing templates, integrated within the production tools; and a focus on cross-platform interoperability, with particular emphasis on heterogeneous client/server configurations. Hopefully, this research effort will bridge the gap between proposed methods for claiming rights within documents and the realities of the production environment.

1. Introduction

1.1 Information Commerce and Rights

The emerging global information infrastructure will bring forth new outlets for information creators and developers and new business models for publishers, distributors, and their customers. But if creativity is to be truly fostered in this Digital Age, the interests and rights of everyone associated with the information must be protected. The exclusive rights of creators and developers as authors of an original expression, afforded by the "copyright clause" of the US Constitution[1] and refined by further legislation[2], must be safeguarded; likewise the rights of the individual to fair use of network-distributed, copyrighted information must be upheld under certain circumstances[3].

This research effort is intended to bridge the gap between proposed methods for claiming rights within documents and the realities of the production and use environments. Our LicensIt™ technology will provide the multimedia creator with secure control over rights claims and permissions at the point of creation and will give the developer of derivative works a convenient, interactive desktop transaction environment for obtaining authorizations from distant rights holders during the production of new work.

1.2 Copyright, Multimedia, and the Net

The concept that there can be one or more owners of certain exclusive rights to creative expressions is a basic tenet of information-based commerce, rooted in the Constitution. It can be argued that electronic distribution of creative works is merely the next evolutionary step in publishing, and the system which has worked well for paper, motion pictures, video and sound recordings should also work for digitized, network-distributed information. Indeed, current copyright law does not require registration or even notification on original works. Creative works are copyrighted as soon as they are expressed in a material form[4], including various forms of computer storage, and the user of those works cannot assume that the owner's rights have been waived — even if the work is in the "public domain."

Never before has information been so accessible, with the capacity to promote

* This research is part of the Networked Multimedia Information Services (NMIS) project, a collaborative effort involving Dartmouth College and Medical School, MIT, and Carnegie Mellon University. The NMIS Project is supported by a grant from the National Science Foundation (NCR-9307548) with support from ARPA (AO-B231), and by cooperative research agreements with IBM Corporation, Inc. and Turner Broadcasting's Turner Educational Services, Inc. (TESI).

economic and scholarly growth. But digitally-distributed creative works present serious challenges to copyright compliance. These works must be accessible to be useful and of value, but in the digital world today, easy access also means vulnerability to unauthorized, perfect reproductions of the work. With today's technology, it is easy for others to create unauthorized derivative works without proper attribution to the originator or paying appropriate royalties, or for others to publish unauthorized modifications under the original authors' name.

Copyright law protects the rights of both the information creator and the user, in the latter case by ensuring that under certain circumstances copyrights may be fairly infringed upon—fair use[3]. It is common for copyright owners in certain media to specify the bounds of fair use for a work; in many cases this is decided in a court of law. But currently there is no common, systematic way for owners of copyright to identify their works in ways which *affect* the end use of their works. Aside from having the creator noticeably brand the work—not an acceptable option for many end uses of graphic, videos, or audio content—there is no way for end users or derivative developers to know if certain rights have been waived, or if royalty payments are required.

Today's user of network-distributed information who wishes to comply with whatever copyright restrictions might be placed on a work will probably need to obtain a license for certain uses. Given a copyrighted work on the network today, no way exists for users of this work to automatically license its use over the network, based upon who they are and their intended use. The possibility of compliance is greatly diminished when usable works are found on the network with no attribution, yet the owner's claims to copyright are no less valid.

A unique problem arises with digitally-distributed audio or video recordings of individuals, whether they are documentary interviews or recordings of performances. They appear in the context of some work which is covered by a copyright, likely held by the producer of the work, but they retain exclusive rights to their "performance" unless these rights have been specifically waived. Usually such performers sign a waiver permitting the producer limited use; this means that certain other uses, by that producer or by derivative developers, may not be permitted. Currently there is no way to bind this kind of permissions hierarchy to the digitized work. This is a particularly critical issue as we work to make health-care-related multimedia resource available on the network.

Finally, it is clear that compliance starts with the proper preparation of the data when the content is first produced—on the desktop of multimedia developers. Currently there are no copyright management systems which are designed to be a integrated part of the multimedia developer's creative process.

1.3 LicensIt™ Goals & Objectives

The primary goal of this work is to create methods and tools which will enable creators of networked multimedia programs to identify their data and claim their rights. We believe this information should be bundled with the data element, and this identification/attribution should persist through generations of derivative use of that data. Our objective with LicensIt will therefore be to demonstrate the application of copyright permissions to a hierarchy of network-distributed data objects to effectively protect owners' rights.

Our second goal is to find a way to facilitate the licensing of multimedia content by different classes of users. We are developing a desktop tool which will be integrated with selected viewing or production tools and which will feature an interactive licensing template; this licensing client will carry on transactions with a remote authorization server. Our objective will be to develop and demonstrate a mechanism for the integrated support of hierarchical permissions headers in the production environment. Further, we will demonstrate the feasibility of networked interactive licensing within the production environment based on hierarchical permissions.

Our final goal is to provide a highly effective solution for networked copyright management which is easily adapted into existing production environments. This implies that it will be easy and convenient to use, with low operational overhead. Our objective will be to show that a cost-effective middle ground exists between the unprotected exchange of data

on one end of the protection spectrum, and blanket encryption (with no value-added copyright features) on the other. We believe our work will help pave the way towards electronic revenue collection for derivative usage of multimedia creations as well as lifetime rights management.

2. Network Interactive Multimedia

Interactive multimedia presentations incorporate creative works of many types, including text, audio, still images and graphics, and motion video and animations. *Please see Figure 1.* Some form of control software or "script" determines how this information is presented to the user, based upon the user's reactions to elements of the presentation already experienced. In highly "linear" multimedia programs the exposure of the user to information is very predictable; in "nonlinear" programs the user is relatively free to map an arbitrary course through the information, and thus the exposure to content is less predictable.

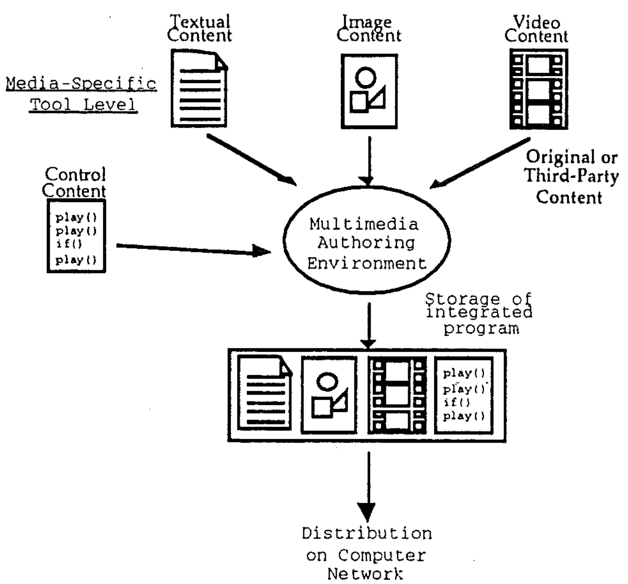


Figure 1 Content Elements in Networked Multimedia

Today's successful interactive multimedia programs are built upon standalone platforms, with the some of the "content" (audio, video, and/or animation) stored in analog format on laserdiscs or in digital format on CD-ROMs. Networked interactive multimedia strives to bring these interactive capabilities to a remote user by way of the network. While excellent work has been done to date with hypertext-based access to networked content[1], as yet these presentations do not demonstrate the same high production values or levels of interactivity as do stand-alone productions. Widespread acceptance of video encoding standards such as MPEG[5] and the ensuing availability of low-cost hardware decoders will soon result in improved production qualities for network-deployed content.

Network-served multimedia presentations are extremely vulnerable to copyright violation. The easy access to multimedia content which the network provides, combined with the unprotected and perfectly reproducible nature of this data, exposes this data to opportunistic copyright infringement.

3. The LicensIt™ System

The LicensIt system is based on a document format which provides a secure container for heterogeneous multimedia data types. This package may encapsulate almost any binary data object; the value of the system is in its ability to restrict 'unwrapping' the package to a controlled environment, specifically from within a LicensIt-ready application or program extension (ie: Plug-in) which can provide the requisite controls over document usage.

LicensIt documents may only be opened and manipulated on LicensIt-ready applications, which can include:

- Stand-alone LicensIt applications, like LicensGIF, a demonstration GIF viewer under development.
- Applications for which LicensIt extensions or plug-ins have been written. Plug-ins for Adobe's PhotoShop and Premiere are planned.
- Applications with integrated LicensIt kernel code. Proposals for integration into Web browsers such as Mosaic or Netscape are being considered.

---

[1] NCSA Mosaic, developed at the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign, was the first wildly popular World Wide Web browser. Others are now in widespread use, including Netscape from Netscape Communications Corporation.

The LicensIt package augments the multimedia data content with supplementary information which fully identifies the source, registry, and format of the data; the copyright legacy of the data; minimal permissions for use of the data as received; and a digital signature which may be used to prove the authenticity of the data. Sufficient information is provided to enable a potential licensee to engage in an online licensing transaction to obtain additional permissions for derivative development or other usage not covered in the minimal permissions set.

3.1 LicensIt™ Document Overview

The basic LicensIt document may contain the following components. These components are examined in greater depth elsewhere[6].

- LicensIt format information
- The LicensIt registration server and document registration codes, including auxilary registration server or identification information
- The bulk data, in an arbitrary flat binary format and (typically) encrypted
- Traceable identification of source works and negotiated permissions used in the creation of the current document
- Permissions of performers whose image or audio likenesses appear in the current document
- A set of minimun permissions which are to be distributed with all authentic copies of the document
- An RSA-based digital signature of the document, facilitating the authentication of the document[7, 8]

Figure 2 provides a schematic view of a LicensIt document.

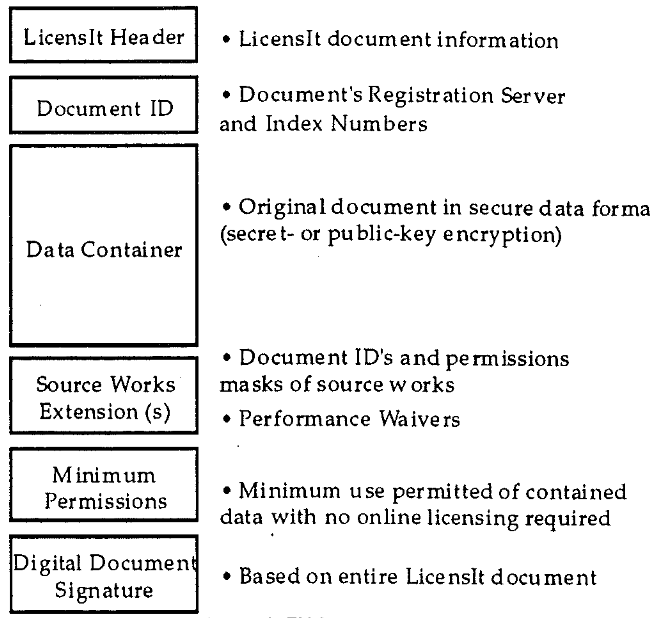


Figure 2 LicensIt™ Document Format

While only the digital signature needs to be encrypted to ensure the authenticity of the document, encryption of the bulk data is recommended, since this is the only way to guarantee that only a LicensIt-ready application can open the file.

Since the LicensIt document format encapsulates document ownership and permissions with encrypted binaries of the work, the rights hierarchy and persistence embodied in this system are applicable to all forms of multimedia content, including the underlying control program. It is clear that this system will enable, within developer applications or rendering tools, a very literal and absolute implementation of the owner's granted permissions for that specific content element or derivative work. In the majority of cases, particularly those in which the derivative works will be commercially available, this literal interpretation is probably desirable. But there may be times when the developer's definition of infringement, supported by a fine-grained rights declaration, are in conflict with a court's interpretation of fair use[2].

Currently there is no clear and absolute way for the creator of a derivative work to be sure if a specific use falls within the bounds of fair use. Typically it is this confusion which leads to infringement litigation. We believe that LicensIt's provision for attaching *minimum*

---

2 The United States Code provides general guidelines for judging fair use [17 U.S.C. 107] which include: the purpose and character of the use, including whether such use is of a commercial nature or is for nonprofit educational purposes; the nature of the copyrighted work; the amount and substantiality of the portion used in relation to the copyrighted work as a whole; the effect of the use upon the potential market for or the value of the copyrighted work.

permissions will encourage creators to grant unlimited local use of their works to developers, while still retaining control over broader (i.e.: commercial) use. Further, for cases of fair use requiring widespread distribution, the creator of an eligible derivative work may obtain a set of auxiliary permissions based on their authenticated digital signature, proving appropriate affiliation.

3.2 System Overview

Figure 3 illustrates how copyright permissions will be integrated into the multimedia production environment. At the top we show individual content elements which have been created with media-specific tools, including text editors (BBEdit®, Word®), graphics tools (PhotoShop®, Debabelizer®), and digital video production tools (Adobe Premier®, Avid Media Composer®). In a conventional production environment these elements would simply enter a multimedia asset library, ready for use in production. No copyright information whatsoever would typically be affixed to the data objects prior to archiving.

In the LicensIt model, content element-specific permissions are affixed to each data object before passing on to the next level of production or on to archiving. Initially our system will use a stand-alone application to affix permissions and other related authorship information to the data. In the long term, it will be more convenient for developers to have such tools integrated into their media-specific tools; to this end we will implement "plug-in" tools for applications like PhotoShop®, Premier®, and Media Composer® based on the LicensIt™ kernel.

The heterogeneous content elements may be released to the production library after the LicensIt encapsulation. During this stage of production a multimedia authoring environment (Kaleida ScriptX®, Apple Media Toolkit®, Macromedia Director®, AimTech IconAuthor®) may be used to create an interactive multimedia program which is a composite of these archived elements. At this point the control characteristics and asset utilization of the program, embodied in the control "script," may also have a permissions header affixed. Thus all of the component assets will be protected in a similar fashion.

This promises to be an effective strategy for managing both in-house and externally obtained assets, but we must look deeper to find a way for these permissions extensions to affect final program integration and execution. For multimedia program integration, a two-tiered rights clearing scheme must be implemented, in which both the encapsulated minimal permissions and the auxiliary permissions[3] of all incorporated works are verified prior to compilation. The specific content of this combination of permissions, including the permissions introduced by the creator of the composite work, will dictate what sort of authorization is required at execution time.

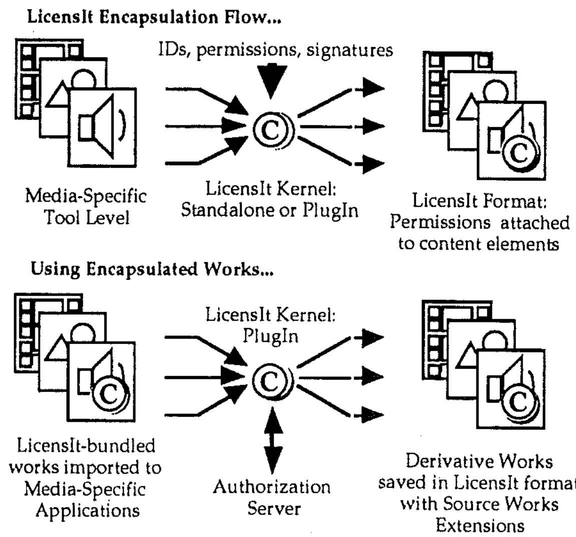


Figure 3 LicensIt Document Flow

Upon remote execution of the compiled multimedia program a spectrum of authorization schemes will be possible, from free execution to the networked authorization of individual assets. We see that the licensing functionality of the LicensIt™ kernel is applicable during execution as well as production. A deliverable for this work will then be a demonstration of how LicensIt™ can be incorporated into the class library of a particular multimedia authoring environment, such as ScriptX®.

---

[3] In this work *auxilary permissions* will be the end result of an on-line licensing transaction—a "receipt" that augments the minimum permissions bundled with the content element.

Key elements of this work include the development of a secure document format and the LicensIt™ kernel. But also significant will be the design of the authorization or license server. Using a simple digital signature scheme for identity authentication, this server will support LicensIt™-based licensing transactions, generating receipts which will be the basis for clearance headers or will serve as keys for execution. This server will understand *classes of users*, enabling different receipts to be issued to different classes of users. This is how we will be able to differentiate between commercial and educational users, for example.

3.5 Summary

This work in progress includes the demonstration of how copyright permissions headers can be applied to a hierarchy of network-distributed data objects to effectively protect creators' rights; the development & demonstration of a mechanism for the integrated support of hierarchical copyright permissions in the production environment, LicensIt™; the demonstration the feasibility of networked interactive licensing within the production environment based on the hierarchical permissions approach; and paving the way towards electronic revenue collection for derivative use of multimedia creations.

At the completion of this work we will be able to demonstrate how electronic copyright management and multimedia authoring can interoperate. *By this effort we will be testing an essential, enabling technology for collaborative networked multimedia development.*

4.0 References

1. U.S. Constitution, Article 1, Section 8, clause 8 (1787)

2. Copyright Act of 1978

3. 17 U.S.C. §107

4. 17 U.S.C. §102(a)

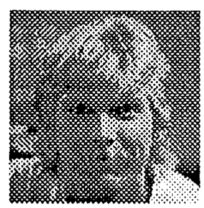
5. D. LeGall, *MPEG: A Video Compression Standard for Multimedia Applications.* Comm. ACM, Apr. 1991. **34**(4): p. 46-58.

6. John S. Erickson, *A Document Format for Secure Copyright Management.* Interactive Media Lab Technical Report JSE-0395, 1995.

7. R.L. Rivest, A. Shamir, and L. Adleman, *A Method for Obtaining Digital Signatures and Public Key Cryptosystems.* Communications of the ACM, February 1977. **21**(2): p. 120-126.

8. Phillip Zimmerman, *PGP™ User's Guide.* Vol. I: Essential Topics. 1994, Phil's Pretty Good Software.

About The Author



John Erickson is a Research Assistant at the Interactive Media Lab, Dartmouth Medical School, and a Ph.D. candidate in Engineering Sciences at the Thayer School of Engineering, Dartmouth College. His research work at IML includes creating practical architectures for the deployment of interactive multimedia educational programs on high- bandwidth computer networks; of particular interest is the development of tools for the protection of copyrights in network- deployed multimedia programs. Since arriving in the Hanover area John has also done consulting work in association with Matrix Simulations of Hanover, NH, involving advanced multimedia applications. From 1984 until 1992 John was a Principal Engineer at Digital Equipment Corporation in Marlboro, Massachusetts, where he was the system architect and project leader for a number of advanced test equipment development efforts. He holds a BSEE (1984) from RPI and an M.Eng (1989) from Cornell University.

The Art of Intellectual Property Strategy

Carey Heckman, Stanford Law School
Co-Director, Stanford Law and Technology Policy Center

Disaster await those who try to exploit every intellectual property right to the maximum extent permitted by law. Getting the most from intellectual property is an art. Today's competitive market demands that you master the art of intellectual property strategy.

Mr. Heckman teaches technology law at Stanford Law School and co-directs the Stanford Law and Technology Policy Center. Before coming to Stanford, Mr. Heckman was at Novell, Inc. (1989-92) as vice president, senior corporate counsel, and assistant secretary. Mr. Heckman had previously been general counsel for Excelan, Inc., which was acquired by Novell in June of 1989. Before joining Excelan, Mr. Heckman was a partner at Ware & Freidenrich (now Gray Cary Ware & Freidenrich) in Palo Alto (1987-89). Mr. Heckman was an associate at Ware & Freidenrich (1983-87) after having been an associate at Morrison & Foerster in San Francisco (1980-83). Mr. Heckman was a law clerk to the Honorable Edward Allen Tamm, Circuit Judge of the United States Court of Appeals for the District of Columbia Circuit.

Mr. Heckman received a J.D., cum laude, from Northwestern University school of Law, where he was Articles Editor for the Northwestern University Law Review, and his A.B., magna cum laude, from Dartmouth College. Mr. Heckman is the author of a number of articles involving computers and high technology. In addition, Mr. Heckman's professional activities include being on the advisory board of the Software Entrepreneurs' Forum (1991-present). Mr. Heckman is also the general chair of the 1995 Computers, Freedom and Privacy Conference.

HTGraph: A New Method for Information Access over the World Wide Web

Yee-Hsiang Chang
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304-1126
email: yhc@hpl.hp.com

Ellis Chi†
Massachusetts Institute of Technology
500 Memorial Drive
Cambridge, MA 02139
email: eyc@mit.edu

† This work was done when Ellis Chi worked at HP Labs in Palo Alto, CA

Abstract

HTGraph (hypertext graph) represents a new information accessing method based on our observations of the current World Wide Web structure. Our method extends the current Web navigation feature, which broadens its scope. The tool also couples database tools with the Web include both the navigation and database accessing paradigms. Further, the tool tries to associate Web information with real-life objects, such as a file directory structure in our case to improve usability.

1. Introduction

The World Wide Web is based on hypertext (or hypermedia). Its structure consists of nodes and links. Nodes can be Web special script files, documents, images, audio clips, and video clips; links connect those nodes over the network. When a user navigates in the Web, he/she views the content in the node. Then he/she can select one of the links for the next destination. Once the destination is reached, he can then view and pick the next one.

This kind of navigation is adequate if the purpose of the access is just looking around in a limited scope. However, it presents a problem when the number of nodes is huge. For example, assume the hypermedia nodes are arranged in a tree structure with layers, and there are ten choices in every node. Once the user selects his choice in the first layer, he/she sees only the ten choices in the selected node and misses 90 choices in other nodes. There will be 1,000 choices in layer three, ten thousand choices in layer four, and so on. In other words, the current Web navigation allows only one path out of many possible paths. The user normally loses the overall perspective once he/she is deep in the Web.

To solve the above problem, the solutions so far employ database technology through various robots [BOWM94; MACB94; MAUL94; DECE94], which automatically collect all information on the Web to build up the database. Database technology has proved its scalability in accessing a vast amount of information, so the solution is valid in this respect. However, database technology requires users to specify the search subject. If users are not exactly sure what the subject is called, database tools are less helpful. Furthermore, this approach takes no advantage of the inherent Web navigation structure, which uses links to associate (or cluster) information among nodes. The current solutions do not collect link information.

Our solution takes advantage of some previous hypertext work [NIEL90; RIVL94] and applies it to the Web environment. Our tool shows a larger scope of the Web (better than any single node can provide) through a graph with all the link information. Similar to the database tools, our tool also explores nodes on the Web and collects their information -- specifically, the title of each node. Unlike the database approach, our tool collects not just nodes but also links among nodes. Furthermore, our tool intends to associate information with real-world objects to facilitate usability, using a hierarchical graph structure similar to the file directory structure. This similarity helps people to browse in a larger scope.

The main contribution described in this paper is that the design utilizes the Web's special features. Specifically, we take advantage of the home pages, which normally represent the starting point for a particular topic. Also, we explore nodes based on the way information is arranged around the home page. In other words, we collect nodes and links starting from these home pages to a specified degree to capture related information before the information diverges to other topics. Our tool, HTGraph (HyperText Graph), has all of the above features.

The rest of the paper is divided into four sections. Section 2

describes the concepts and operation of HTGraph. Section 3 surveys related work in this area. Section 4 shows the HTGraph's data structure and algorithms for node exploration. In section 5, the graph layout design is discussed.

2. HTGraph

2.1 Concept

We have three observations on the current hypermedia-based Web. First, information is normally organized or started from home pages, where one home page is equivalent to the root of a file directory. The home page serves as an entry point for a set of detailed nodes for a particular topic or institution that that home page represents. So, having as many home pages as possible at the first level of browsing is very useful. Here, we define a home page as being located at the highest layer. Any node that can be reached from the home page is considered as being at a lower layer; the exact layer depends on how many links are between the node and the home page.

Second, when a user follows the links on the hypermedia, he/she can take only one path at a time. The user loses more overall perspective the further he/she goes down through the layers. So, we would like to show all these choices at the same time.

Third, as the layer gets lower (y-axis in Figure 1), the topics get wider (x-axis in Figure 1).† The content and usefulness of information diverges and some is completely unrelated to the initial home page's topic. In fact, some of the lower layer nodes are the home pages of other topics. We would like to extract only the nodes in which most of the information is related to the topic of the initial home page.††

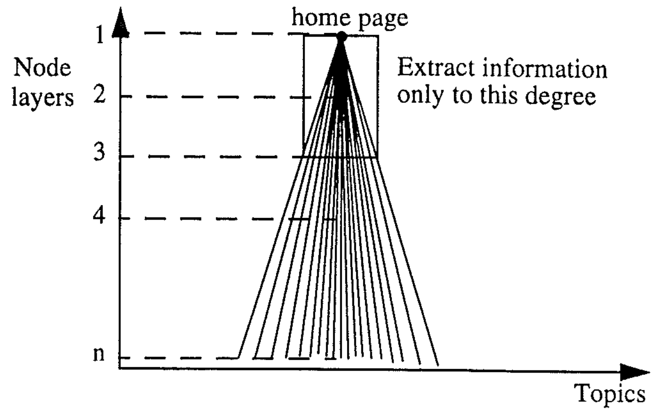


Figure 1. The Divergence of Topics Further Down the Node Layers

HTGraph is a tool to show the relationship among Web nodes as a directed graph by taking the above observations into consideration. The relationship among nodes is shown by displaying either all hypermedia references starting from the home page and its descendants, or the nodes that represent many physical linkages hiding inside the nodes -- that is, logical grouping. A good example of logical grouping would be a graph that hides linkages of a node if it has too many hypermedia nodes. Figure 2 shows an example of logical grouping.

Unlike the database approach, which retrieves only the nodes, we also collect and show the links among nodes. By doing so, we maintain the relationship among these hypermedia nodes and keep the Web's navigation feature. These links also represent natural groupings of similar nodes. For example, the MIT home page contains all the linkages to its related Web nodes. It currently contains some cultural events information in Boston for newcomers to MIT. Using the database technology, a user might not be able to specify the right query and retrieve such information. In our approach, these nodes are already linked together and are retrieved together.

† Figure 1 shows only a linear increase in the number of nodes as a user moves down the layers. In reality, this increase can be exponential, as stated earlier.

†† How much we should explore to capture the most information is unknown. Actually, each home page and its links are arranged differently depending on the creator of the page. We don't expect that a common number for the degree of detail will apply to all home pages. Currently, in HTGraph, a user can select the number of nodes he/she wants to explore.

Note that the solution further allows a "space jump" in the Web even without the exact address or Universal Resource Locator (URL) of a node. In other words, instead of following the existing node links by clicking at nodes one after another, users can "space jump" to a particular node by clicking the node on the graph.

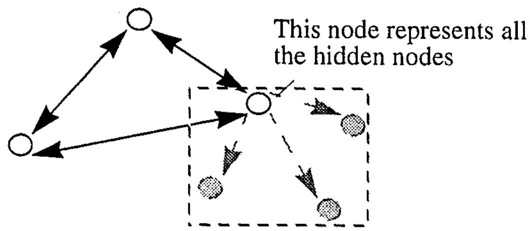


Figure 2. Logical Grouping

2.2 Operation

To run HTGraph, the steps are as follows:

Step 1: Use the existing database tool to generate the starting view.

We suggest to use either of two existing database tools, Harvest [BOWM94] or Lycos [MAUL94], to do the generation of a starting view. A user first uses the selected database tool to collect the home pages for the starting view. There are different starting views depending on the user's preference. For example, if displaying all the possible "http://www.x.y.z" URLs as the starting point is a good idea, the user uses the tool (Lycos or Harvest) to retrieve the nodes that fit this pattern. Another preference could be all the "http://www.x.y.**edu**" URLs for the educational starting points or "http://www.x.y.**com**" URLs for the commercial case.

Step 2: Explore the selected home pages based on the specified degree.

The user runs the exploration part of HTGraph to access nodes from each home page to the specified degree. The tool automatically builds up the link and node information. This process should be done off-line (e.g., midnight every day) as is the case for the database and step 1. However, if the user has concerns about whether the information is current, step 2 can be skipped, and the HTGraph exploration will be done in step 4. The trade-off is performance, since the latter case requires doing both exploration and display at the same time.

Step 3: Generate the initial display with all the selected high level nodes.

The initial display of the starting view is generated when a user starts the display part of the HTGraph program with the selected starting view. We can associate the initial home pages with some physical objects. One example, shown in Figure 3, is to associate the nodes with a map. When a user starts to browse and wants to check on the Web sites in the San Francisco Bay Area, he/she double-clicks the node on the map, and a blow-up screen shows all the home pages in this region. This part is currently done manually and is still under development.

Step 4: View each individual home page graph by clicking on the page.

When a user clicks on http://www.hpl.hp.com in Figure 3, a graph is shown for this home page and associated links and nodes. Figure 4 shows the blow-up of http://www.hpl.hp.com using HTGraph for the first ten nodes, which is the degree we specified. Note that the nodes are shown similar to a directory structure. Also note that we display only the titles of the nodes instead of their addresses because the titles contain more meaning about the nodes. Since some of the titles are very long, users see the full title only when the cursor is on the specific node. Figure 4 shows an example: when a user accesses this home page, he/she can see right away that there is a node with the title "Management Profile" in layer three. He/she can then decide whether to access this node or not.

Step 5: View each node by clicking on the node.

If the user is interested in any node, he/she can click on the button and the document will be displayed. This is what we referred to as the "space jump" earlier, because a user can see further down through the layers from the HTGraph display and access the node directly.

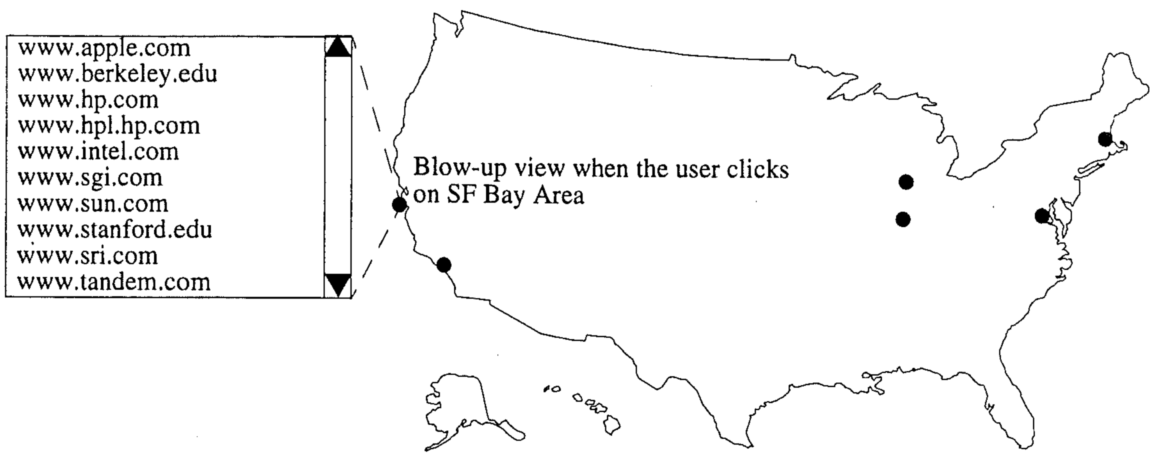


Figure 3. An Example of a Starting View from the HTGraph

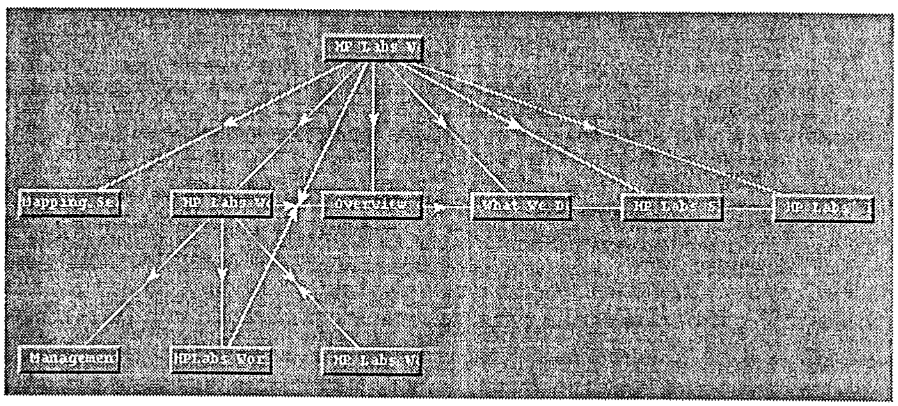


Figure 4. Display of a Wider View Starting from a Home Page

3. Related Work

3.1 Hypertext

Navigation over the hypertext technology has been studied extensively over the years [MARC88; MARS89; NIEL90; CREE91; RIVL94]. The fundamental issues have been identified and various solutions proposed. In [RIVL94], navigation over the hypertext is assisted by a structure point of view of the overall system. The user interface issue also has been addressed by [MARS88] using multiple windows, and by [NIEL90] using maps. We are taking advantage of the ideas from these efforts to apply to the Web environment.

3.2 Navigation vs. Database

One of the authors has investigated various information accessing methods in [CHAN94]. These methods fall into two categories: navigation and database. The database technique has been used extensively in the business environment. The technology is scalable in terms of the ability to access a vast amount of data. However, it is less helpful when a user has little idea about what he/she wants to see and wants only to look around, as is the case for most TV viewers. On the other hand, the navigation paradigm has proved to be powerful for the broadcast world. "Channel surfing" is a simple form of navigation within the broadcasting environment. The World Wide Web creates another form of navigation through the hypermedia links on the Internet.

3.3 Robots and Databases

Many robots have been developed on the Web [BOWM94; MACB94; MAUL94; DECE94]. Their primary purpose is to add a database function into the Web environment. The World Wide Web Worm [MACB94] and spiders [DECE94] represent tools to explore the Web and retrieve information. There are also *archie* to search the *ftp* space and *veronica*

for the *gopher* space [DECE94].

The Lycos [MAUL94] and Harvest [BOWM94] tools built on the Worm's techniques, are the two most recent tools to couple Web information with current databases. They also improve the Worm technology by collecting information in a more efficient manner.

4. HTGraph Data Structure and Algorithms

Our implementation allows input of the degree of exploration and collects not just the node but also the link information. The data structure responsible for building HTGraph is called Node. Node has the following data structure:

```
struct _node {

        /* first part: info needed to build HTGraph */
        HTAnchor * anchor;
        struct _node * next;
        char * heading;
        C_list * FirstChild;

        /* second part: info for printing HTGraph */
        BOOL Printed;
        int XCoord;
        int YCoord;
};
```

A node contains two major parts. The first part keeps information for making HTGraph. The second part is responsible for printing the graph. Nodes are linked into a link list, headed by HTGraphLink (or FirstNode). The tool explores all the nodes by performing a breadth-first search. There are three major issues in building HTGraph:

- Node exploration
- Linkage to hypertext nodes
- Loop avoidance

4.1 Node Exploration

Exploring a node means to search through the whole hypermedia node, get all the accessible hypermedia references, and establish linkages. Since there is no particular goal (i.e., a particular hypermedia node) for the search, our consideration narrows down to depth-first searches and breadth-first searches. A depth-first search is out of the question, since the depth of the search may be infinite, in which case the exploration degenerates into merely retrieving the first hypermedia references in all explored hypermedia nodes. Therefore, a breadth-first search should be the most appropriate for HTGraph node exploration.

The breadth-first search algorithm presented here is slightly different from the one that is usually found in a textbook. Since the exploration here is not searching for a particular node, the way to stop the search would be either by setting up a time-out or by specifying a limit on the number of nodes explored. An ordinary breadth-first search does not care how nodes are related to one another, so it removes a node whenever it is explored. However, in HTGraph, all the explored and unexplored nodes are queued in HTGraphLink, and further linkage is implemented to relate parent and child nodes. Below is an ordinary BF algorithm and the BF algorithm for HTGraph; Figure 5 shows how HTGraphLink looks after node P is explored. Queuehead points to the node P that is currently under exploration. In the example, P is found to have three hypermedia references (called "children"), C1, C2, and C3. These children are appended at the end of the queue, since they have not been visited. After this, the queuehead moves to the next item on the link.

An ordinary BF search†

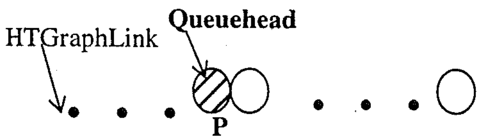
- Form a one-element queue consisting of a zero-length path that contains only the root node.
- Until the first path in the queue terminates at the goal node or the queue is empty,
    - Remove the first path from the queue; create new paths by extending the first path to all the neighbors of the terminal node.
    - Reject all new paths with loops.
    - Add the new paths, if any, to the back of the queue.
- If the goal node is found, announce success; otherwise, announce failure.

A customized BF search

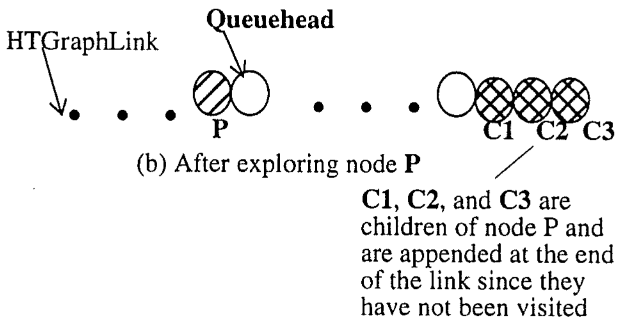
- Put a home hypermedia node in the BF queue.
- Until time-out or the queue is empty or specified degree is reached,
    - Explore the first unexplored node from the queue.
    - Check for loops (see whether the node has been visited (ChildVisited())).
    - Add the unexplored nodes, if any, to the

† Extracted from Patrick H. Winston, *Artificial Intelligence*, third edition.

back of the queue.



(a) Before exploring node **P**



(b) After exploring node **P**

Figure 5. Node Exploration of Node P

4.2 Linkage

We need a field that is responsible for keeping track of a node's hypermedia references or children. Since the number of children is not the same in each node, a structure with variable size is needed. A link list is chosen. By using the field FirstChild, which points to a link list that consists of a structure that points to the child, the parent-and-child relationship is established. Figure 6 shows how the linkage works using FirstChild. In the example, there are three children belonging to the first node. The first pointer on the child node points back to HTGraphLink. If the child has been visited and been recorded on the link, it points back to the location. Otherwise, it points to the new location that is created at the end of the link. The second pointer on the child node points to the next child node (NextChild) on the list.

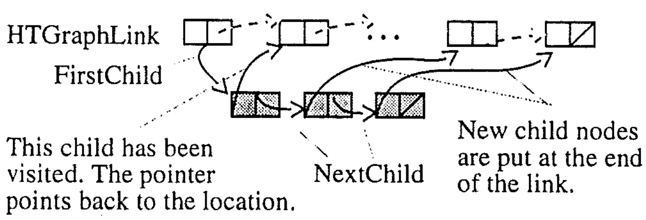


Figure 6. Child Linkage Using FirstChild

4.3 Loop Avoidance

Loop avoidance tackles the following problems:

[1] Looping.
This is a common phenomenon. An example would be a home page having its hypermedia reference referring back to itself (as shown in Figure 7). So when a graph is built, we have to make sure that a hypermedia node is not explored more than once. Otherwise, the exploration would be an endless loop between two nodes.

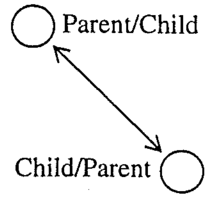


Figure 7. A Two-Node Case

[2] Multiple-parent nodes.
It is very common to find a hypermedia node being referred to in several hypertext nodes. We call this hypertext reference a multiple-parent node. A multiple-parent node should be printed only once in the graph; it should be displayed as a node being pointed to by a couple of nodes.

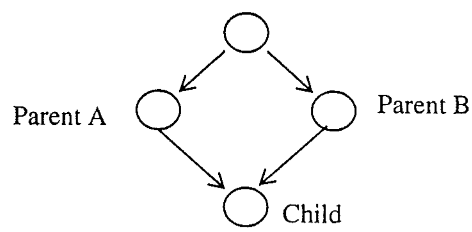


Figure 8. A Multi-Parent Case Where Parent A Shares a Child with Parent B

To ensure that a node is explored only once, the program makes sure that an explored hypermedia node will not be queued for a BF search again. To do that, a procedure named ChildVisited() is used to check if a hypermedia node has been explored (i.e., to check if the node is already in HTGraphLink). If the node has been explored, ChildVisited() returns a pointer to the node in HTGraphLink. If the node has not explored, ChildVisited() puts the node at the end of the BF queue and updates HistoryList. HistoryList is used to record the URLs of all explored hypermedia nodes and pointers to those nodes in HTGraphLink. Simply speaking, ChildVisited() gets the URL of the inspected node, compares the URL with the explored nodes' URLs, and decides if it should return the pointer to an existing node or create a new node and put it at the end of the queue. The structure of HistoryList is

shown in Figure 9. HistoryList is an array that distinguishes addresses based on their length. Addresses that have the same length are linked together in a link list. The data structure in the link list consists of three fields. The first field is the address, used to determine whether this address has been visited. The second field is a pointer that points to the location of the node in HTGraphLink, and the third field is a pointer that points to the next record. Note that the data structure for HistoryList is not optimized.

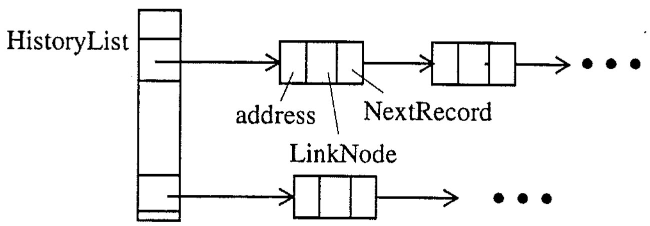


Figure 9. Data Structure of HistoryList

5. Node layout for HTGraph

To lay-out the nodes and links among one another is the key to the popularity of the tool. The difficulty in displaying a good graph is that there are many loops and links, which can occupy the same space. Currently, we are still seeking the best layout algorithm, one that can minimize the crossing of links and improve the legibility of HTGraph.

We have considered two ways to lay-out the graph. First, we can start the home page in the middle of the display, and then print all its descendants around the home page. Second, we can print the graph as a tree. The first method makes the display closer to a map, but it is hard to assign an empty spot for a hypermedia node. The second method, on the other hand, is easier to implement and shows a sense of hierarchy, which is chosen.

To make a printout of the graph, the HTGraph program generates *tcl/tk* [OUST94] command lines for each explored node in a script file that is concatenated to another script file containing the definition of the commands. The final script is invoked and the display is shown on the canvas in *tcl/tk*'s wish command. Figure 10 shows the printout of the two-node case and the multi-parent case.

6. Conclusion

In this paper, we identify the problem of accessing a vast amount of information in today's World Wide Web. We point out that the database solution takes no advantage of link information and is not very useful when the user has little idea about what to look for. We then propose a new method for Web navigation, which has resulted in a tool called HTGraph. This tool uses the features from the Web. It takes advantage of the home pages to collect nodes surrounding these home pages; it shows links among nodes in the graph, which offers natural indications about the relationship among nodes; and it also associates information with real-life objects, such as the file directory structure in our case, to improve usability. Moreover, our tool is scalable in terms of showing various levels of detail. It also allows the user to perform a "space jump" directly to the destination.

References

[BOWM94] Bowman, C. M., Danzig, P. B., Hardy, D. R., Manber, U. and Schwartz, M. "The Harvest Information Discovery and Access System," *Proceedings of the Second International World Wide Web Conference*, Chicago, Illinois, October 1994, pp. 763-771.

[CHAN94] Chang, Y. H., "Wide Area Information

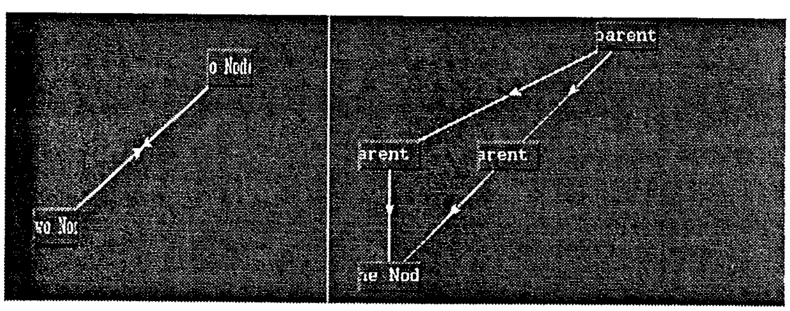


Figure 10. Two-Node Case and Multi-Parent Case.

Accesses and the Information Gateways," *Proceedings of the 1994 1st International Workshop on Community Networking*, July 1994, pp. 21-27.

[CREE91] Creech, M. L., Freeze, D. F., and Griss, M. L., "Using Hypertext in Selecting Reusable Software Components," In *Proceedings of the Hypertext '91 Conference*, December 1991, pp. 25-38.

[MACB94] McBryan, O. A., "GENVL and WWWW: Tools for Taming the Web," *Proceedings of the First International World Wide Web Conference*, May 1994.

[MARC88] Marchionini, G., and Shneiderman, B., "Finding Facts and Browsing Knowledge in Hypertext Systems," *IEEE Computer*, 1988, pp. 70-80.

[MARS89] Marshall, C. C., "Guided Tours and Online Presentations: How Authors Make Existing Hypertext Intelligible for Readers," In *Proceedings of the Hypertext '89 Conference*, 1989, pp. 15-26.

[MAUL94] Mauldin, M. L., and Leavitt, J., "Web-Agent Related Research at the CMT," *Proceedings of the ACM Special Interest Group on Networked Information Discovery and Retrieval*, August 1994.

[NIEL90] Nielsen, J., *Hypertext and Hypermedia*, Academic Press, San Diego, California, 1990.

[OUST94] Ousterhout, J. K., *Tcl and Tk Toolkit,* Addison-Wesley, 1994.

[RIVL94] Rivlin, E., Botafogo, R., and Shneiderman, B., "Navigating in Hyperspace: Designing a Structure-Based Toolbox," *Communications of the ACM*, vol. 37, no. 2, February 1994, pp. 87-96.

[DECE94] December, J., "New Spiders Roam the Web," *Computer-Mediated Communication Magazine,* 1(5), Sep. 1, 1994.

A System to Facilitate Teaching and Learning with Network-based Interactive Multimedia

Daniel C. O'Connor
Interactive Media Lab
Dartmouth Medical School
and
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755
doconnor@dartmouth.edu

ABSTRACT: This paper will discuss the general requirements for success for a network-connected computer-based learning system. It will then describe a system currently under development at the Interactive Media Laboratory (IML) to enable the use of network-based interactive multimedia in all educational settings. This system will enable technical and non-technical educators to create interactive learning modules incorporating network-based media sources. The system also provides the student with an adaptive learning module navigation and use tool.

1. Introduction

The use of computers in the classroom has been predicted by many as one of the potential saviors of public education (Schank & Cleary, 1994). The computer is an enabling technology that must be combined with specialized software and access to information to become an educational tool. Interactive multimedia learning systems [the specialized software] with network connectivity [the access to information] are seen as a way of improving individual instruction in the face of growing classrooms and shrinking resources (NRENAISSANCE Committee, 1994).

"When you have that wonderful combination of an interested student ... and a gifted teacher, there's no replacement for that. The problem is that's a very rare event" (Henderson, 1992). Networked interactive multimedia has the potential to provide every student access to gifted teachers. As Schank (1994) puts it, "Why not have a collection of Nobel Prize winners as your own personal physics teachers?"

There are many factors that will determine whether this prediction becomes reality or not. In this analysis, we consider an entire learning system – from the teacher to the students and back. We start by considering some of the key factors that stem from the most important viewpoint, that of the student.

1.1. To structure or not to structure

Studies have shown that while using an interactive multimedia learning system, simply providing an abundance of time and hypermedia links does not guarantee learning for students that are exploring a topic for the first time (Mayes, Kibby, & Anderson, 1990; Recker, 1994). Students who are unfamiliar with the topic under study have no way of knowing what facts are important or even what questions are relevant. Blattner (1994) calls these students "lost in hyperspace." In the same article, she points out the need for "the user [to be able to] freely explore the data space, yet to provide guidance in the process. This proves particularly important if the user is unsure of the choices." Structure is not an automatic panacea, however. Recker & Pirolli (1992) showed a positive interaction between a student's advance in ability and increased performance using a less structured method of presentation.

These findings make the case that a successful learning system should be able to adjust the amount of structure based on the student's perceived or measured ability. Yet, for practical reasons, the creator cannot be required to produce multiple versions of one learning module to accommodate this adaptability.

1.2. Watch, listen, or read?

An effective learning system must be adaptable beyond the degree of structure. Given alternative methods of explanation (and with network access to a large number of multimedia databases, why shouldn't they?), the learning system must be able to decide which of the provided methods is best to be shown first for the current student. The factors in this decision include the teacher's ranking of the general effectiveness of each of the alternatives, the student's preference of methods (if any), and which methods have caused the student to best comprehend similar topics in the past. This decision is not exclusive. After viewing the first alternative, the student can choose to use other methods. Also, the learning system could provide various methods for remedial purposes.

It is important to note that the method of presentation is not equivalent to the media through which it is presented. For example, two alternative methods for emphasizing the possible results of drunken driving are: 1) showing the end result of many alcohol-caused automobile accidents; or 2) interviewing one teenager who was the driver and sole-survivor of one alcohol-caused automobile accident. While both of these are video, their methods are far from equivalent.

1.3. Assuming facts not in evidence

The background of the student must be taken into consideration when presenting a new learning module. The student must not be presented with new information that relies on capabilities not yet acquired. Similar to Dannenberg & Joseph (1992), the learning system must be able to modify the learning module to omit foundation-less material, if not critical to what the student is trying to learn, or to direct the student to the appropriate learning module that contains the necessary background material, if available. For example, if the student is in a module on atomic structure, it is unnecessary to know about strong and weak forces if the goal is to learn about covalent bonding. It would be necessary, however, if the student was investigating basic fission processes.

1.4. Collaboration

Collaboration is essential for success in learning. The more opportunities for collaboration made available to the student and the greater the effectiveness of those opportunities, the better a student's educational experience. There are many types of individuals with whom a student may collaborate:

1) another student, whether that student is in the same classroom or across the country.

2) a live teacher, a real person tasked with the success of this student's learning experience.

3) a virtual teacher, a combination of the learning system and learning module which can answer some general and a few specifically asked questions.

4) a live expert, a real person recognized as knowledgeable in the field of the learning module who has volunteered or is being paid to cooperate with the students.

5) a virtual expert, like the Ask system (Schank, 1994), which can guide the student to answers and also to better questions.

All of these collaborations can occur either synchronously (i.e. real-time) or asynchronously (i.e. via enhanced email). Synchronous collaboration can either be via text/graphics, voice, videoconference, or face-to-face. Isaacs & Tang (1994) describes the advantages of videoconference over voice or text collaboration and its disadvantages when compared to face-to-face meetings.

Asynchronous collaboration should be conducted via enhanced email built into the learning system. The enhancements include the ability to capture the context of and the path through the learning module that generated the student's query, etc. Responses to queries should be able to automatically guide the student through the learning module (or other material within the system) with text/graphics, audio, or video annotation to and through the previous point of question.

Two modes of collaboration deserve special attention: synchronous student-to-student and synchronous teacher-to-student(s) collaboration. If a number of students are working on a goal-oriented learning module, as suggested by Schank (1994), a mechanism for agreeing on a course of action must be established. If a teacher is collaborating one-on-one or one-to-many, a means for guiding each of the student's modules through a sequence at the teacher's control is necessary.

1.5. Learning module creation

For a learning module (and system) to be successful, it must accomplish three basic tasks vis-à-vis the student: the module/system must *attract*, *engage*, and *retain* the student. To attract the student, the module must be easy to use, robustly designed and must attempt to warm the student to the subject quickly. For the topic to be truly learned, as opposed to memorized for regurgitation on a test, the student must be engaged not only intellectually but emotionally as well. Retaining the student requires the system to be responsive, the module to grow with the student's increased understanding, and the system and module both to present as high a quality presentation as the sources make possible.

The actual creation of a specific learning module would approximately follow these steps. First, a topic or topics to be covered would be decided upon and the scope and depth of coverage set. The determination of the prerequisite background knowledge for each topic and depth within each topic would then be determined. Before the widespread proliferation of learning modules, the prerequisites may be

specified to be completion of a traditional course or a pretest.

The creator of the module would then begin an iterative process that includes determining potential methods of presentation for each of the topics, envisioning the sources that would satisfy those methods, and searching for the appropriate sources that are actually available. The major requirement for any potential method is that it "tell a good story" (Schank, 1994), the material presented should be appropriate to the audience and should compel the students to want to learn more. Even if the module initially contains only one method of presentation, more may be added as inspired by the discovery of new source material or through student feedback.

The search for available source material will be manual (and thus extremely limited) until the widespread use of agents (Ramanathan & Rangan, 1994) enables automatic searches through available databases for appropriate material. Even with agents, it is likely that there will be discrepancies between the desired and available sources. The creator of the module then must decide to either create missing sources or to alter the methods of presentation. After the list of methods is finalized and the sources are collected, the creator moves into the last stage, composition.

Above all, the composition environment must accommodate non-technical creators as well as it does technical ones. All composition must be able to be completed without the need for a programming-like language. In many instances (public primary education, e.g.), a learning module creator does not have the time to become a programmer to get the job done. After the location of source material, creating a basic lecture-style learning module should take no more time than would be required to create a traditional lecture. The composition system should allow for fine grain control over the module without requiring it. Finally, the composition system must be extensible. It must be easy to create module styles and to distribute them to other creators.

1.6. Requirements Summary

In this introduction, we discussed several factors that may determine the success or failure of a networked interactive multimedia learning system. The system must be able to adapt the structure of a learning module to the ability and previous experience of the student. The method of presentation must be dynamic, based on available sources and the preferences of the teacher and student. The module must be able to be dynamically altered depending on the background of the student. The system must enable various modes of collaboration among students, teachers, and experts. Finally, the module creation process must be accessible to non-technical creators and the system and module must *attract*, *engage*, and *retain* the student to be successful.

2. The IML Teaching and Learning System

The Teaching and Learning System (TALS) currently under development at IML will be based on ScriptX[1]. It will be developed to run on both Apple Macintosh and IBM-PC compatible personal computers. TALS assumes that the computer is capable of full-screen, full-speed[2] MPEG-I digital compressed video and audio, displaying both video and graphics on the same screen with basic overlay capabilities. Synchronous collaboration will require either sound input or video and sound input. During the development of TALS, the use of CUSeeMe[3], MovieTalk[4], and nv/vat[5] will be explored for their applicability to this environment.

TALS consists of two major components: the Teaching Tool and the Learning Tool. Each of the components is described in the following sections.

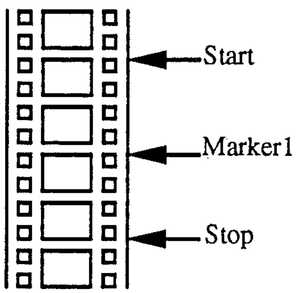
2.1. TALS Teaching Tool

The Teaching Tool has three major sections: the source compilation, module composition, and collaboration utilities.

The source compilation utility contains a completely graphical World Wide Web browser that requires no HTML programming. This Web browser will interface with the composition utility allowing for the selection of network-resident text, images, sounds, and movies. The compilation utility will have simple source viewers that will allow for

[1]ScriptX is a cross-platform multimedia development platform from Kaleida Labs, Inc.
[2]full-screen, full-speed is 640 pixels by 480 pixels, 16 bits per pixel (minimum), 30 frames per second, with CD quality sound output.
[3]CUSeeMe is an Internet-based videoconferencing tool from Cornell University.
[4]MovieTalk is a QuickTime component from Apple Computer for adding videoconferencing to a QuickTime-capable computer with video input.
[5]nv/vat are the MBone network video and visual audio tools for UNIX/Xwindow workstations.

the definition of desired start, stop, and synchronization points (sometime called markers) without altering the original material. Synchronization points allow source objects to be synchronized in the program, for example coordinating images with a narration or providing subtitles for a movie. This definition will result in the creation of data structures similar to edit decision lists (EDLs) in video editing. Figure 1 shows an example of an edit decision list for a Web-resident MPEG file.



URL = //www.here.edu/~me/example.mpg
Start = 00:01:10:15
Marker1 = 00:01:23:07
Stop = 00:02:15:00

Figure 1: Network EDL Example

The composition utility has to accommodate the wide variation in structure and content that is assumed in the basic learning module format. It also has to be easy to use, yet allow for fine-grained control of the composition. To accomplish this, each piece of source will be treated in an object-oriented fashion. Each content object will be accessible by appropriate methods and will possess class and instance variables (Arbab, Herman, & Reynolds, 1993; Hardman, Bulterman, & van Rossum, 1993a; Hardman, van Rossum, & Bulterman, 1993b; Herman, Reynolds, & Davy, 1993). Variables will include spatial and temporal composition values as well as start and stop transitions, if any.

A hybrid of absolute and relative temporal specification will be used. In absolute temporal specification, the start and stop times of every content object are specified in relation to a single clock source. This method is also known as timeline composition[6]. In relative temporal specification, the start and stop times of content objects are specified in relation to the start, stop, and synchronization times of other objects (Little & Ghafoor, 1993; Little, Ghafoor, Chen, Chang, & Berra, 1991; Prabhakaran & Raghavan, 1994). Figure 2 shows an example of hybrid temporal composition.

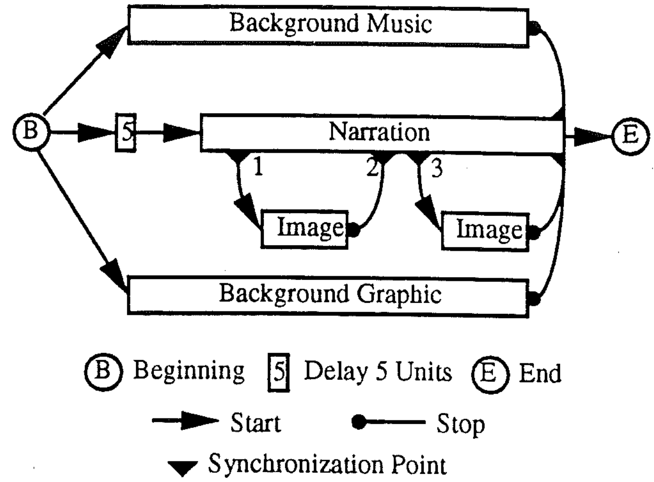


Figure 2: Hybrid Temporal Composition Example

At the beginning of this simple example, background music starts to play and a background graphic is shown. Five time units later a narration begins. When the narration reaches the first creator-defined synchronization point, an image is shown. At the second synchronization point, the image is removed. At the third synchronization point, another image is shown. At the end of the narration, the second image and the background graphic are removed and the background music stops playing.

The composition utility will support MPEG-, QuickTime-, and AVI-encoded video and audio and PICT, DIB, and JPEG still images and SND-, WAV-, and µlaw-encoded audio. An appropriate standard for formatted text is still an open issue. Two features that are planned to be added in the long term are import/export of HyTime and MHEG documents.

The other major feature of the composition utility is the ability of the creator to specify the quality of a module as perceived by the student. This includes the quality of the images, sound, and video as well as the responsiveness of the system. Of course, these quality figures cannot be greater than the quality of the source material or the capabilities of the students' computer. Quality figures of merit are shown in Table 1.

[6]timeline-style composition is typified by Adobe Premier.

Medium	Figures of Merit
Video	frame/second, bits/pixel, resolution, size, Q
Image	bits/pixel, resolution, size, Q
Audio	sample/second, bits/sample, mono vs. stereo
Response	time delay from user selection to result

Table 1: Quality Figures of Merit

The collaboration utility will support enhanced email through appropriate MIME extensions and synchronous collaboration as discussed above.

2.2. TALS Learning Tool

The Learning Tool has three major sections: a Web browser, a collaboration, and a learning/tutoring utility. The Web browser is a subset of the Web browser in the Teaching Tool without the interface to the compilation tool. The collaboration utility is the same as the Teaching Tool collaboration utility.

The major development effort in the Learning Tool section of TALS is the learning/tutoring utility. The learning part of this utility is fairly straightforward. It must be able to navigate a learning module in all its various configurations. The learning utility must decide whether to access the learning module's source material from the network when needed or to cache all or part of the sources to local storage before the student uses the learning module. The quality requirements of the module, the bandwidth available between the student's computer and the locations of the source content used in the module, and the capabilities of the student's computer affect this decision.

The tutoring part of the learning/tutoring utility is one of the greatest development challenges in the entire system. It is assumed that its features will grow and become more refined as the project continues. As described in the introduction, there are three major areas in which the tutor must make decisions that affect the student's interaction with the system: background checking, structure selection, and method of presentation selection.

The tutor needs to keep track of the knowledge/ability background of the current student so that no information is presented that is completely outside of the student's ken.

The first time a student uses a learning module, the tutor would select the most structured approach. The more a student uses a particular module and shows increasing knowledge or ability, the less structured the presentation will be.

The largest challenge for the tutor will be selecting among various methods of presentation for a given student. Topics covered in learning modules and various methods of presentation will have to be categorized and the student's success using various method/topic combinations will have to be rated and stored. For students new to TALS, the tutor will rely on the recommendations of the learning module creator and then on the preferences of the student.

2.3. Project Summary

TALS is a development project to create a cross-platform network-based interactive multimedia learning system which adapts to the students that are using it and facilitates the creation of high quality interactive multimedia by educators without requiring programming knowledge.

3. Bibliography

Arbab, F., Herman, I., & Reynolds, G. J. (1993). An Object Model for Multimedia Programming (No. CS-R9327). Computer Science, Department of Algorithmics and Architecture, Centrum voor Wiskunde en Informatica.

Blattner, M. M. (1994). In Our Image: Interface Design in the 1990s. IEEE Multimedia, 1(1), 25-36.

Dannenberg, R. B., & Joseph, R. L. (1992). Human-Computer Interaction in the Piano Tutor. In Multimedia Interface Design, M. Blattner & R. Dannenberg (Eds.), (pp. 65-78). Reading, MA: Addison-Wesley.

Hardman, L., Bulterman, D. C. A., & van Rossum, G. (1993a). The Amsterdam hypermedia model: extending hypertext to support *real* multimedia (No. CS-R9306). Computer Science, Department of Algorithmics and Architecture, Centrum voor Wiskunde en Informatica.

Hardman, L., van Rossum, G., & Bulterman, D. C. A. (1993b). Structured multimedia authoring (No. CS-R9304). Computer Science, Department of Algorithmics and Architecture, Centrum voor Wiskunde en Informatica.

Henderson, J. V. (1992). Innovations: The Future Is Now-Part 3. WNET: New York City, NY.

Herman, I., Reynolds, G. J., & Davy, J. (1993). MADE: A Multimedia Application Development Environment (No. CS-R9360). Computer Science, Department of Algorithmics and Architecture, Centrum voor Wiskunde en Informatica.

Isaacs, E. A., & Tang, J. C. (1994). What video can and cannot do for collaboration: a case study. Multimedia Systems, 2(2), 63-73.

Little, T. D. C., & Ghafoor, A. (1993). Interval-Based Conceptual Models for Time-Dependent Multimedia Data. IEEE Trans. on Knowledge and Data Engineering (Special Issue: Multimedia Information Systems), 5(4), 551-563.

Little, T. D. C., Ghafoor, A., Chen, C. Y. R., Chang, C. S., & Berra, P. B. (1991). Multimedia Synchronization. IEEE Data Engineering Bulletin, 14(3), 26-35.

Mayes, T., Kibby, M., & Anderson, T. (1990). Learning about learning from hypertext. In Designing Hypermedia for Learning, D. Jonassen & H. Mandl (Eds.), (pp. 227-250). Berlin: Springer Verlag.

NRENAISSANCE Committee, Computer Science and Telecommunications Board, Commission on Physical Sciences Mathematics and Applications, & Council, N. R. (Eds.). (1994). Realizing the Information Future: The Internet and Beyond. Washington, DC: National Academy Press.

Prabhakaran, B., & Raghavan, S. V. (1994). Synchronization models for multimedia presentation with user participation. Multimedia Systems, 2(2), 53-62.

Ramanathan, S., & Rangan, P. V. (1994). Architectures for Personalized Multimedia. IEEE Multimedia, 1(1), 37-46.

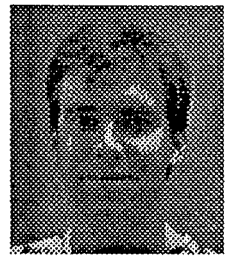
Recker, M., & Pirolli, P. (1992). Student strategies for learning programming from a computational environment. In Second International Conference on Intelligent Tutoring Systems, (pp. 382-394). Berlin: Springer Verlag.

Recker, M. M. (1994). A Methodology for Analyzing Students' Interactions within Educational Hypertext. In ED-MEDIA, Educational Multimedia and Hypermedia Annual, 1994 . Vancouver, B.C., Canada.

Schank, R., & Cleary, C. (1994). Engines for Education. Evanston, IL: Institute for Learning Studies, Northwestern University.

Schank, R. C. (1994). Active Learning through Multimedia. IEEE Multimedia, 1(1), 69-78.

About the Author



Daniel O'Connor is a Research Assistant at the Interactive Media Lab at the Dartmouth Medical School and a Ph.D. candidate in Engineering Sciences at the Thayer School of Engineering at Dartmouth College. He is also a computer systems consultant with Archetype Engineering Company. From 1988 to 1992 he worked at Apple Computer, Inc. as an I/O systems architect and designer. Dan graduated from Dartmouth College in 1988 with an A.B. in Engineering Sciences. His current research interests include network delivery of interactive multimedia and investigating novel scalable architectures for high performance, low cost hypermedia authoring and presentation systems.

The author can be contacted at:

Interactive Media Lab
Dartmouth Medical School
7275 Butler 1
Hanover, NH 03755
Phone: 603-650-1363
Fax: 603-650-1164
Email: *doconnor@dartmouth.edu*

Dynamic Authoring and Retrieval of Textual Information: DARTEXT

Albert K. Henning
Associate Professor
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755-8000
al.henning@dartmouth.edu
http://hypatia.dartmouth.edu/henning/henning.html

Mimi Jett
Chief Executive Officer
Electronic Technical Publishing
2906 N.E. Glisan Street
Portland, OR 97232
mimi@teleport.com

***Abstract*-The super-exponential growth in the base of information creators and users with access to the Internet makes possible a variety of schemes for the creation, organization, dissemination and revision of information over the Internet. In this work, the ramifications of this technology for academic publishing, particularly in the engineering sciences, are explored. Frameworks are proposed which enable and encourage dynamic authoring and retrieval of information that, in the past, would have been associated with a textbook. A case study of the concept's application to an undergraduate course in engineering systems analysis is presented.**

I. Introduction

In the span of a very few years, we have crossed the watershed of information production and delivery using the technological bridges of the Internet, and ubiquitous information browsers such as Mosaic. The paths leading outward from this new shore are innumerable. Here, we describe one path for creating, organizing, disseminating and revising information, using these bridges and their present and future companions. While the path is general, we will discuss it in the particular context of university-level science and engineering education, and give a specific example in terms of the topic of *Engineering Systems* at Dartmouth College.

Conventional science and engineering courses in colleges and universities revolve around a set of lectures, homeworks, exams, laboratory exercises, and usually a textbook. Frequently, though less so of late, the textbook is the prime focus of the course. Lectures tend to follow the table of contents of the textbook. Homework is assigned verbatim from it. Instructors rely on published solutions to the homework.

Despite some past success, there are considerable problems with this model of instruction. While some students and faculty appreciate the permanence of a textbook, others find it constricting. Cost increases in textbooks have far outstripped inflation over the past twenty years, leading to text prices in the sciences and engineering in the range of $100. Too many students purchase a textbook, use it briefly during a quarter or semester, then sell the book, typically back to the bookstore from which it was purchased.

There is another drawback to the conventional means of authoring and publishing textbooks, which hinders the process of teaching and learning. Contemporary authors exercise an unfortunate tendency to exert conscious, complete control over the content and presentation of their writing. Typically, an author will insist on a specific, linear order of presentation of material, with each section fixed and immutable. We refer to this insistence as the "Outer Limits Syndrome": the desire to "control the vertical and the horizontal", forcing the reader to assume a passive posture in the learning process.

The intrinsic hubris of this tendency stems from two fallacies. The first fallacy is that one author, or even several authors, can 'know it all'. The majority of teachers (and students) know this to be false. Even in the most tradition-bound institutions or courses, teachers will add supplemental notes, or depart regularly from the sequence of a textbook. The reasons for these departures vary, but include a desire to establish personal control over the course material. A need to adjust the textbook to suit the local curriculum may also dictate departures. Or, an instructor may wish to prevent students from succumbing to the othodoxy of a textbook, thus losing the edge of critical and independent thinking.

The second fallacy is that students are empty, passive slates, upon which the author writes with the chalk of knowledge. Again, students and teachers know this assumption is untrue. Formal student feedback concerning the quality of every aspect of a course of instruction is gathered at the end of many courses in the sciences and engineering across the country. These evaluations become useful in improving a course, and assisting its positive evolution. In every aspect *except* the textbook, changes can be wrought in time to create improvements for the next presentation of the course. Textbooks, however, must await completion of the process of producing a revised edition. Publishers decide prior to printing of the first edition, what the revision cycle will be: two, four, or six years in length. Frequently, even in this instance, revised editions suffer from little substantive, direct feedback from the most intimate users of the material, the students themselves.

Education research over the past decade has demonstrated traditional methods of instruction presume a single mode of teaching and learning. Professors lecture, assign homework, give written

exams, ensure work is graded, and assign course grades based on a curve. Students take notes, execute solutions to closed-form problems, study, and take written exams, largely in isolation. Contrary to these patterns, progressive educators attempt to address the diverse learning modes of their students, rather than demand all students adjust their learning patterns to the professor's singular mode of instruction. Open-ended problems and laboratory exercises, group projects, collaborative homework, untimed exams, and course grades based on an absolute scale (as opposed to a curve), constitute some of the techniques currently employed.

We are attempting to incorporate these insights into a new means for the creation, dissemination, and revision of academic, textual information. However, by no means have our ideas been conceived *ab initio*. Some specific, successful attempts to correct deficiencies in teaching and learning have influenced our thinking, and deserve mention here.

Mook [Hen94] has undertaken significant reforms in the teaching of introductory physics at Dartmouth College. A key attribute of his efforts unlocks student frustration in a unique way. Students from previous classes are employed to create problems and solutions, supplementary notes, lab modules, videos, and multimedia displays which address and clarify issues these same students found difficult or confounding. The impacts are profound. The student developers are empowered to learn and communicate in new ways, and their efforts result in improved learning and teaching for subsequent classes.

Mazur [Maz91a, Maz91b] has also conceived and implemented introductory physics reforms at Harvard. He has completely changed his lecture style and format. His lectures now revolve around what he calls ConcepTests. Each one-hour lecture is broken into four segments. In each segment, a particular concept receives focus. Mazur first discusses the concept, in some detail, and occasionally with a brief example. A relatively simple, multiple choice question is then posed to the class, based on this concept. Students are first asked to think about the question, and frame their answer. They are then asked to enter their answers, on either a machine-readable card, or into a digital device which keeps statistics on student responses throughout the lecture, and throughout the course. Next, students work in pairs to discuss the problem and their individual approaches, and arrive at a common ground. Finally, the students record their answers, changed or unchanged, once more. ConcepTests succeed as a teaching and learning tool, and (since statistics are gathered) the success is measurable. The explosion of sound during the pair discussions is less measurable, but still powerful. It brings an intimacy previously thought to be impossible for a large, introductory class lecture setting.

The *Primus* system from McGraw-Hill was a publishing environment intended to enable greater flexibility in the organization and presentation of textbook information. Other publishers attempted similar projects. The central idea was to allow instructors to create their own textbook for a particular course, by selecting chapters from the 'stable' of book titles managed by a specific publisher. The market has largely rejected these products, for a variety of reasons. The price/performance ratio for these products was generally too high. Though the cost to students was in the $25-50 range, the quality of the product -- styles were uniform but very plain, colors were limited to black and white, and binding was paperback or soft-bound -- was insufficient to overcome the lower price. Instructors felt constrained by the limited number of titles held by a publisher, and by the restriction that whole chapters only from each title chosen must be used. Publishers also expected other publishing houses to collaborate, and submit material from their own lists for inclusion. When this participation did not occur to the extent predicted, the idea began to fade.

Redish [Red93, Red94] has led the University of Maryland's efforts in revolutionizing introductory physics education. The use of the computer is a principal component of this effort. The broad-based approach (of looking at a wide-range of systems which physical principles can describe) is similar to that taken at Dartmouth in the context of Engineering Systems (whose case study is described below). Such an approach can be facilitated and enhanced by the use of the Internet and related tools.

Mathematics instruction at Duke University, specifically calculus instruction, is being treated as a laboratory science [Moo92]. Calculus is no longer merely an esoteric exercise, but is coupled intimately to its original source in 'natural philosophy'. The interactivity thus wrought has broken the limiting bonds of traditional introductory calculus teaching.

A multimedia development workshop for engineering faculty will be given for the first time during the summer of 1995 [Har95]. Funded by NSF, the workshop endeavors to make authoring of multimedia, academically related works relatively simple, and to disseminate this information in substantive ways.

Few attempts have yet been made to use the new technological bridges to effect dramatic and constructive change. Some notable exceptions exist, though even these have shortcomings. Larson's work [Lar94] discusses the construction of an interactive calculus textbook. Strict control of content by the authors is implicit, even in this interactive work. Larson emphasizes correctly that graphic design is frequently a time consuming task. Shortcuts cannot be made in graphic design, without compromising impact and, ultimately, success. Proofreading is also a time-consuming task, according to Larson, which has been

given little consideration by developers of hypermedia information sources. The shortcomings in Larson's approach will be addressed in subsequent sections.

Aminmansour [Ami94] has also made inroads on some of the problems we identify here. The interactive multimedia book on steel design places great emphasis on graphical interface quality, and on interactivity. Important provisions are also made to solicit and incorporate feedback from student and faculty users of the database (or 'software', in the language used by this author).

The Global Network Academy [GNA94] has taken some first steps toward publishing texts, and organizing their presentation and structure. The flexible input concepts contained in their documentation parallel some of the approaches detailed herein.

Our concept is somewhat similar to Aminmansour's, but goes further. As in [Lar94] and [Ami94], we begin with a focused database of textual information. Our emphasis is on academic subjects, and subjects (such as VLSI Design) which lend themselves to technical training. Without question, however, our concept may be extended to other arenas, since the database *content* lies at its core. And, regardless of the specific content, each database must be dynamic, living, and breathing.

The database must be flexible enough to include information in any form. Text, sound, still photos and graphics, animated or moving pictures, may all be incorporated.

To facilitate our concept, users of the database must have simple means to suggest changes and improvements, and well-satisfied expectations that their suggestions will be incorporated. Just as in a technical journal, the graphical interface -- the 'look and feel' of the database -- must be well-planned and extremely consistent. Its specifications must be public, with ample access to translators between many different formats, to allow virtually anyone to author contributions and revisions using their favored composition environment.

Retrieval of database information must be simple and low-cost. This necessity is already well-facilitated. Most academic environments have ubiquitous connections to the Internet. Many require students to purchase personal computers. Most other institutions will follow suit in the near future, as the cost of even mid-range computers with the necessary performance drops to attainable levels.

Authoring and retrieval of information, therefore, are the keys which unlock the door to the center of our concept. And it is the *content* of the information database which constitutes the core. For us, this information lies in the realm of academic science and engineering. However, our concept is completely general, and can be extended to other realms of information.

Content is our focus, but it must be supported strongly by other frameworks. As much as possible, we seek to build on the positive aspects of the Internet and the World Wide Web. At the same time, we must preserve the necessary roles filled today by publishers, textbook authors, production sub-contractors, and others vital to the textbook publication industry. And, we must add new players to the sphere of activity, to leverage new features and power made possible by evolving technology. These attributes are discussed more thoroughly in a subsequent section.

Control over the information in the database is essential to our concept. However, such control must be exercised carefully, delicately and elegantly. Too much control, and our concept becomes no better than current textbooks. Those attractive and powerful features -- interactivity, universal access, and rapid incorporation of new or revised material -- available through the Internet will be lost. Too little control, however, and anarchy will take hold, leading to an unattractive and ultimately unsuccessful product.

We intend for a professional editorial review board to have oversight responsibility for each database. This board will be similar to the review boards of most professional technical journals. It will, however, have special responsibility for the overall framework of the database. Furthermore, review board members will have a financial stake in the database, and be contributors to its content. Rapid turnaround times, between submission of new or revised information, and its incorporation into the database, must be a hallmark of the review board.

To clarify the path we envision, we have broken down our overall concept into smaller, interrelated frameworks. These are presented in the next section.

II. FRAMEWORKS

Our overall concept is depicted in Figure One. At the heart of our concept lies the *content*. We conceive of four principal frameworks in support of the content, which are at once linked intimately with the content, and each other. These are:

- Administration
- Graphics
- Intellectual Property Transactions
- Financial Property Transactions

Note that it is not necessary for all of these frameworks to reside under the umbrella of a single company. Though these frameworks constitute activities long managed by traditional publishers, in fact, it will be desirable for each framework to be owned by a small, agile firm, with support from several existing publishing houses.

In the following sections, we describe in more detail these individual frameworks. Following this discussion, we will present an example of the content for one possible database, based on Engineering Systems Analysis and Design.

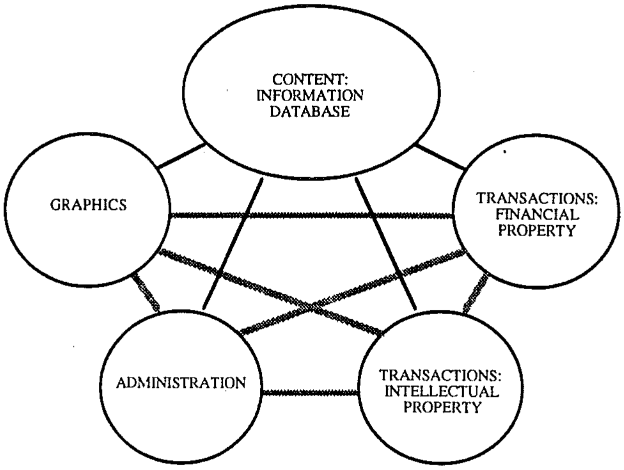


Figure One: The DARTEXT Concept

A. Administration

Figure Two depicts the administrative framework which supports the content-focused information database. This framework has several specific functions, which are listed below. This trend is consistent with the present-day 're-engineering' of the American corporation, where the responsibility for individual corporate functions are being spun off to independent companies.

Marketing, Sales, and Distribution: For any particular database to be truly successful, it must produce revenues which exceed costs. This assumption implies the need for these functions. In and of themselves, they do not differ from their counterparts in traditional publishing. However, to promote our concept, these functions must incorporate the new technology based on the Internet in order to advertise, sell, and distribute a particular database in softcopy form. Since the database, or major portions of it, will also be realized in compact disk (hardcopy) form, traditional routes of marketing, sales, and distribution must be maintained.

File, Hardware, and Network Services: The database must be maintained in appropriate ways. Its integrity must be preserved and ensured. Appropriate access for authors and retrievers must be authorized.

Physical Production: The creation of hard copies of the database, or portions of it, in paper or compact disk form, must be administered.

Acquisitions: New authors for existing databases must be sought out, and encouraged to make contributions. Authors for new databases must also be sought out. Market surveys and analyses, to determine which new databases are economically viable and should be pursued, must be made.

Legal Services: This area includes torts related to all aspects of the database, except for those agreements concerning intellectual property.

Accounting: As a matter of course, accounting will be a necessary function for all the frameworks supporting the database.

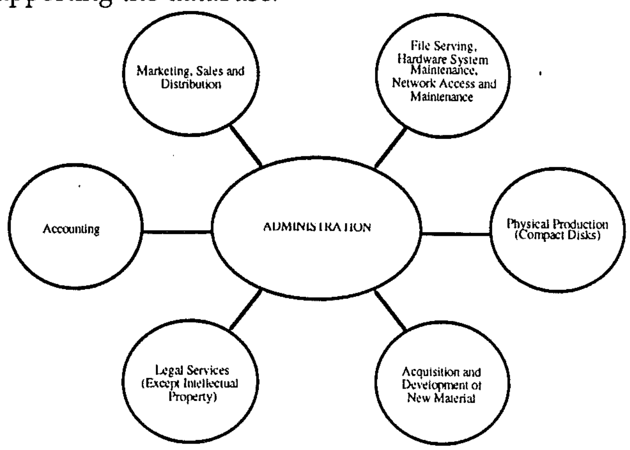


Figure Two: DARTEXT Administrative Framework

B. Graphic Design

Figure Three depicts the graphics framework for the database. We are presuming that the dominant interaction will be visual (by sight) and mechanical (by a mouse or keyboard). However, there is no reason for other means of interaction (e.g. sound) to be excluded. For our purposes here, we refer to the process of presenting the database content as 'graphic design', though perhaps 'interface design' would be a more general and enduring term.

This framework fulfills the following functions:

Database Format Specifications: These specifications refer to the 'look and feel' of the database, as perceived by browsers. It is important for these specifications to be widely available and understood, so that the broadest spectrum of potential and actual authors may be encouraged to submit material for use in the database.

Format Translators and Converters: The database must not constrain authors to use a particular, limited and limiting set of authoring tools. Nor may the database be constrained to be viewed by only a few browsing tools. Just as graphics conversion software, such as GIFConverter [Mit94], allows files of many formats to be read in, and output files of many forms to be generated, so too must the database accept input from a variety of formats, and support browsing using a wide variety of tools. It is likely that the database will exist in a single format (e.g. SGML or HTML), common to most browsing tools.

Authoring and Production Environment: While no single authoring environment can or should become an inflexible standard for all database authors, it is still reasonable for the database to recommend the authoring or multimedia production environment which would streamline the process of bringing new information into the database. As new authoring and

production tools become available (e.g. ScriptX [Kal94], WebFORCE [SGI95], or works being developed at Dartmouth [O'Co95]), they will be assessed for use in the authoring environment.

Browsing and Playback Environment: Some consideration will also be made for making the database compatible with currently available and popular browsing and playback tools. Activity here will center on ensuring the database format can be accessed and presented easily by these tools.

Synthesis with New Tools: The Internet and related technologies are, today, in a state of great flux. The database must, therefore guard against being left behind by the information marketplace, by continual evaluation of new tools, beyond those used in authoring and retrieving information.

Search Engines and Other Tools: Conventional search engines are already employed in a wide variety of Web-accessible documents and information resources. Searches across the Internet are also available. Texis [Mne94] is a search engine available from Mnemotrix, Inc. which has been used largely for non-science applications over the past fifteen years. However, its ability to format a database for searching based on concepts and relationships, rather than simply on keywords, makes it best suited for our purposes.

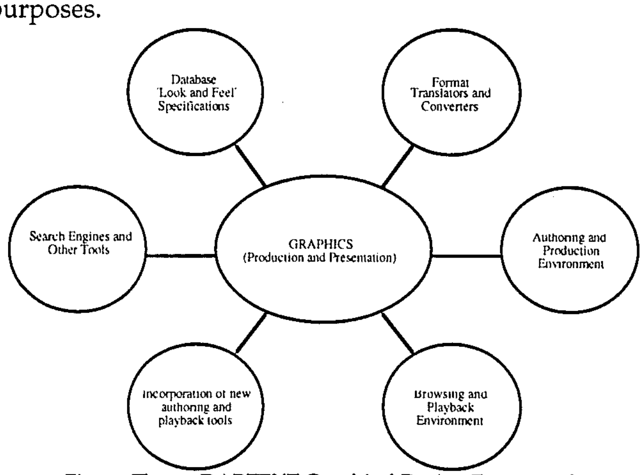


Figure Three: DARTEXT Graphical Design Framework

C. Transactions: Intellectual Property

The essential transaction associated with our concept is the exchange of intellectual property for financial property, and vice versa.

Intellectual property is created by authors, and becomes the database content. Customers access this property in a variety of ways. Since the core of the database is its content, the intellectual property framework is arguably the most important, and is shown in Figure Four.

Content Organization Specification: Within this framework, the organization of the database content must be specified, much the way a textbook is organized according to a Table of Contents, List of Figures, List of Tables, and Index. Here, though, the organization into regular entities will not be as simple as creating textbook 'chapters', with perhaps homework problems and other references listed at the end of each chapter. The place in the database for labs, exams, solutions, video demonstrations and simulations, synthesized software tools, and other new entities must be determined well in advance. Early thinking regarding the framework of the database will reap great future benefits. The database must be flexible to accommodate future submissions, while meeting the needs of present-day instructors and students.

Review of Submitted Material: We expect material submitted to the database will come from a wide variety of authors. Students of all ages and abilities, instructors at all levels, and others can be expected to become, not just consumers of information in the database, but authors and creators of information. Lowering the barriers to submission, by allowing contributions of the longest or shortest lengths to be submitted, is a critical feature of our concept.

We expect any author will use those authoring tools which are most convenient or well-known. We also expect translators between output formats for these different tools will be cheap, reliable, and ubiquitous. In a very real sense, then, many of the production tasks now handled by publishers and their subcontractors will be taken up by the authors, and their software tools, themselves.

Submission of material will take place electronically. The database will have well-publicized standards for the format of submissions, much the way academic journals have standards for font size, typeface, margins, and other attributes of printed material.

An editorial review board will examine material submitted to the database. It shall determine whether the new material fits into the database framework, and if so whether the material should take its own place in the database, or replace existing material. The board also has the responsibility to determine the database structure and organization itself, and make changes to them as changing conditions warrant.

Update Database: Any dynamic object must change in order to improve and survive. It will be the responsibility of the editorial review board to be the 'change agent'. As a consequence, some information, over time, will become obsolete. The board will determine whether newly submitted material is unique, and should enter the database on its own merits; or, whether it re-states material currently in the database in a new way. The board must be willing to take risks and experiment. They must devote a portion of the database to new formats for presentation, or new content, or even new media (for instance, adding sound and video to present textbooks; or adding smells to future textual products).

Legal Services: Legal services in this arena will focus on copyrights and licensing issues. A number of endeavors are underway to address copyright issues for the Internet [Eri94].

Accounting: Again, accounting will be necessary to manage the flow of information into the database, and the exchange of financial property (e.g. shares in the database) for it.

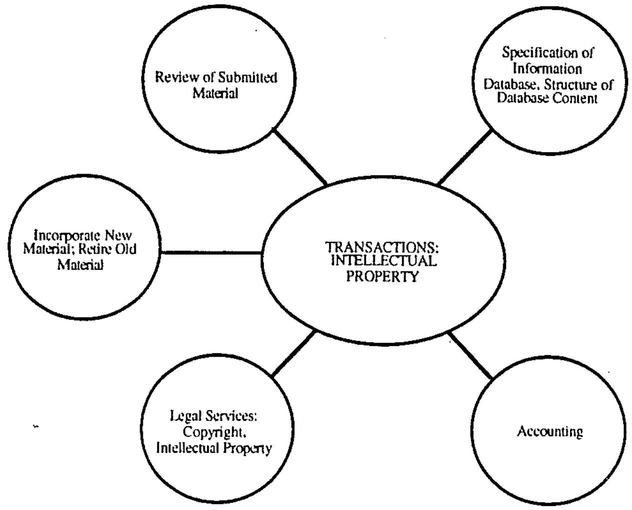


Figure Four: DARTEXT Intellectual Property Transaction Framework

D. Transactions: Finances

Once the intellectual property is created and made accessible, it must be exchanged for some other property, typically financial. Figure Five depicts the support framework for these exchange activities.

Initially, in developing our concept, we considered whether each database should be so self-regulating, that its cost would be free to the individual user. In the end, however, it became clear that each database, to be truly successful, must be operated on a for-profit basis. This conclusion was driven primarily by the realization we had chosen an intermediate level of editorial oversight, between the constricting control exercised by authors of present-day textbooks, and the virtual absence of control exercised over its content by an Internet topical newsgroup. Having chosen a middle path (with intellectual rigidity and high profits on the one hand, and intellectual chaos and no profits on the other), we developed the intention that each database exist on a for-profit basis.

Having made this decision, several options present themselves. Charges could be made on a per-transaction basis. In this scenario, a record of each piece of information accessed by the user would have to be kept, with appropriate charges made for this access, and billings sent on a periodic basis. In some sense, such an accounting scheme would build on, or be parallel to, efforts to implement interactive learning [Eri95]. We believe this choice to be too cumbersome for the maintainers of the database, and too confusing for consumers.

We prefer instead the subscriber model. Users will pay a fixed fee to purchase a compact disk containing the most current version of the database. Our preliminary studies indicate the consumer cost for this CD will be approximately $25. Users will also get access to server for limited period of time. Time extensions and/or CD upgrades may be purchased for a small, periodic fee (e.g. $5/year). Current research into distributed learning environments and their ramifications should solve difficulties which may arise from having database information on both a local hardcopy (CD), and a remote, more updated softcopy. However, our studies indicate customers prefer to receive a tangible asset in return for money, making the CD the most attractive vehicle for distribution of the database.

Authors must receive remuneration for their contributions to the database. In order to handle this necessity, we have conceived of a *stock model* to represent the intellectual capitalization of the database. The database will receive an initial capitalization of, say, ten thousand shares. Following conception and publication of the first structure and organization of the database, the editorial review board will determine how many shares of the total capitalization to make available to authors. Some shares will be held in reserve for future authors and contributors. Contributors whose works are accepted for inclusion in the database will receive shares in return. Profits after expenses, based on subscriptions, will then be distributed to the authors on a per-share basis. The number of shares held by each contributor will be set by the editorial review board, in proportion to the value of the contribution to the database. Once the initial capitalization is exhausted, it will be up to the board to determine when material must be retired (and its contributors must give up their shares), or when new stock should be issued, should the growth and use of the database warrant increased capitalization.

Since contributors of material of nearly any length can receive shares for their contributions, we expect the 'activation energy' for authorship to be small. Potential authors will not be daunted by the need to commit thousands of hours to complete an entire text. Furthermore, they will be encouraged by the knowledge that even small contributions can receive financial recognition.

Purchasing of Services: A variety of services will need to be purchased, from marketing to network hardware. Sub-contracts will be granted as needed to obtain services not available from personal directly working on the database. In general, most services will be obtained via sub-contract, given the current thrust of American business toward more numerous, smaller, leaner, and quicker companies.

Legal Services: Legal services here will be related to memorializing royalties, shares, and other remunerative issues.

Accounting: Share accounting and other accounting related to financial property is covered here.

Share Assignment: The editorial review board will determine the assignment, reassignment, or retirement of database shares to and for authors.

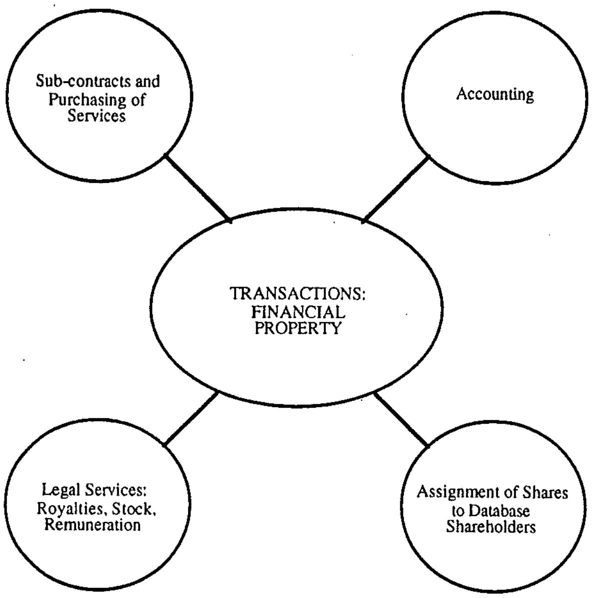


Figure Five: DARTEXT Financial Property Transaction Framework

III. Uniqueness

We believe our concept is unique in important and compelling ways. It mirrors many of the forces and trends in American and world-wide business today: trends toward decentralization, the breakup of conglomerates, 'lean and mean' organizations, down-sized organizations, the spin-off from core competancies of corporate service organizations, just-in-time inventory, and empowering all employees to take greater responsibility for their products. It applies these ideas to the realm of publishing, and creates a new paradigm for publishing in the academic arena. The paradigm is a departure from the idea of a single, or a few, authors as intellectual creators of a text, having complete control over the material. A structural and organizational framework for the textual content of the database takes the place of the textbook, with responsibility for creating material falling to those who will choose to shoulder it, in return for tangible remuneration. In essence, our concept is midway between the rigid form of traditional publishing, and the *hive mind* or *swarm system* conceived by Kelly [Kel94], whose principal attribute is non-controllability, in return for the 'immortality' of the hive or swarm.

Our concept goes beyond recent advances in electronic publishing, which remain at root an exercise of total author control over how the user interacts with an informational database, and even more control over what that database's content may be. Distributed authorship means distributed publication and proofreading costs. It means all users will benefit from a wider variety of viewpoints. Users' feedback on revising and improving the database will be virtually instantaneous compared to conventional textbook publication. The resulting work will not be immutable. Consumers will interact with the product, and have the opportunity to influence its improvement or change.

A. Attributes

Our concept has a number of important attributes. Authorship is distributed. By setting design standards, in the manner that most technical journals these days set layout and design standards, allows authors to create content to these standards, thus minimizing costs of publication. The financial incentives for authorship, even on a small level, promotes active learning by offering incentives for activity, and disincentives to passivity. Students will therefore be more likely to be engaged by the material, through a feeling of authorship beyond mere understanding.

B. Challenges

A number of challenges appear on our horizon. Some must be surmounted before our path can be deemed a success. Most obviously, success will be measured by market acceptance. These challenges are discussed separately in the following paragraphs.

Distributed, Distance, and Interactive Learning: Using communications systems to access information from remote locations has led to the development of systems which facilitate and monitor distributed learning. It is conceivable that tools for distributed learning could also be used to handle copyright management automatically. These tools could monitor use of particular portions of the database, providing the editorial review board with metrics to determine which portions of the database are: unclear and in need of refinement; widely used and appreciated, necessitating perhaps a share readjustment for their authors; or little used, and therefore in need of removal. Interactivity may also be facilitated. In the long run, the database could serve, not only as a source of textual information, but as an evaluative framework for student work. This last extrapolation holds especially for quantitative work with specific answers, such as most present-day homework and exam problems. Open-ended problems, or true design work, will not lend itself to such evaluative services.

Portability: Students have a need for their textbooks to be portable. Studying rarely occurs in a single locale. Lightweight, notebook computers may help extend our concept to address these issues more simply,

though even with present technology, the cost of notebooks is usually too great, while they are neither as rugged nor as portable as a conventional textbook.

Bundling Tools: Many courses in mathematics, physics, chemistry, and engineering employ calculation tools such as Matlab, Mathematica, and Maple as key components of instruction. It is possible such tools, and examples based on them, may be bundled with the database. Such use could potentially lower the cost to the students of the software tools. Their incorporation would, however, complicate the legal and accounting pressures on the database owners, however.

Search Engine: The choice of a search engine for the database is important. We have settled on the use of Texis [Mne94]. Its features for content-based searches, or searches based on lexical relationships between words and phrases, is quite strong, and as databases grow will be essential.

Building Coalitions of Authors and Users: It is critical that the need for the database be agreed upon by parties from a number of institutions, academic or otherwise. Whether the common nature of the institutions drives the collaboration, or their common interest, is somewhat immaterial. In either case, a database conceived or formed by a single person or institution will be likely to fail the test of economic success, which is to be profitable. Toward this end, we have established collaborative relationships with Bucknell University.

Choosing Appropriate Databases: To achieve critical mass, get over the initial activation energy, and propel this concept forward, databases with large audiences should be sought first. Introductory calculus, introductory physics, introductory chemistry, and engineering systems appear to be ideal candidates.

Rate of Information Creation and Annihilation (Turnover): We expect databases to focus on two ends of the information 'frequency spectrum'. Here, by 'frequency' we mean the rate at which new information enters the database, especially new knowledge and original content, and not simply re-statements of older presentations. High frequency databases will therefore focus on leading edge technologies, in the early stages of formation. Low frequency databases will focus on introductory material at the undergraduate level. As an example, once formed we would expect a database focused on introductory calculus to change only slowly. However, a database focused on micro-machines and microelectromechanical systems (MEMS) will change rapidly, as new information in this field is being created daily.

Competitive Databases: If our concept proves successful, we expect competitive databases to arise. Since one of our motivations is to lower the cost of instruction materials for the consumer, this eventuality can only improve the cost and performance of each database product. The electronic 'publisher', or consortium responsible for each database, will need to take competitive positions similar to those employed by present-day textbook publishers. For instance, nearly every academic textbook publisher in the sciences carries one or more introductory calculus titles on their list. We expect no different a result with our concept.

IV. CASE STUDY: ENGINEERING SYSTEMS

We present here a brief example derived from the engineering curriculum at Dartmouth College. *Engineering Sciences 22: Systems* is a course founded on the mathematics of ordinary differential equations. Its goals are to instill in students a systems-oriented approach to the analysis and design of systems of any sort, whose state changes over time, and to learn and use the mathematical tools to execute such design and analysis.

The schematic for the thought process which forms the foundation of this Systems course is shown in Figure Six. Students begin with a real-world system of any complexity. The system can be measured using a variety of experimental techniques, which are also taught in the course. The dynamic behavior of the system is then modeled by, first, creating a simplified conceptual model of the system; second, extracting a mathematical model from the conceptual model, using appropriate physical laws; third, solving the mathematical model using appropriate techniques; and finally, comparing the predicted and measured response of the system. If discrepancies are found which are unacceptable, the cycle must be repeated, with adjustments made at any of the points in the circle.

The figure below becomes image mapped, and serves as the point of departure. Students may click on any of the circular points in the process, to determine more detail about specific aspects of the process. Subsequent figures are also image mapped. [In the HTML version of this document, only certain of the image maps are enabled, for purposes of demonstration.]

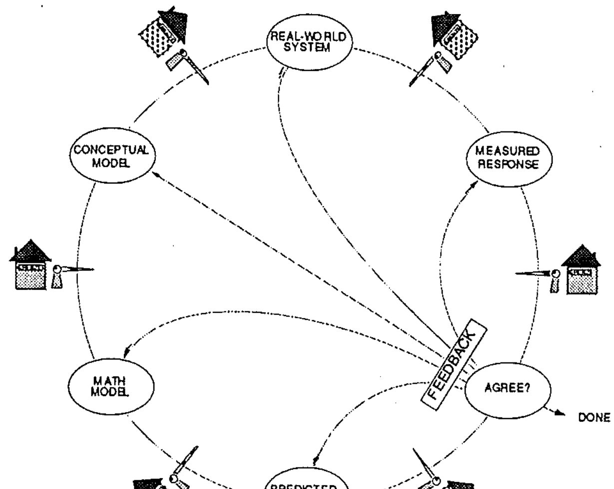


Figure Six: Process Flow Page for Study of Engineering Systems

Let us assume we are a student, given a real pendulum as a system to analyze. Let us also assume the student has already conducted a series of experiments to measure the system's response subject to a delta-function input, has stored the data, and now must analyze the system. The first step is to create a conceptual model. The student turns to the Conceptual Model Page (Figure Seven).

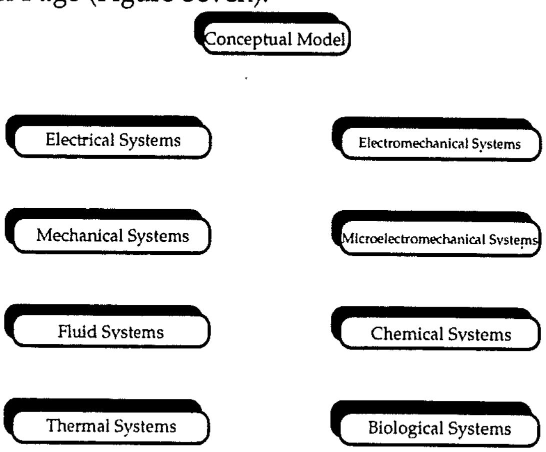


Figure Seven: Conceptual Model Page

Here, there are any number of systems to choose from, but clearly this particular system is a mechanical system. There are a number of avenues away from the 'Mechanical System' hyperlink. Some point toward specific examples, as in Figure Eight, which fit the framework of dynamical systems described by first- and second-order, linear differential equations. Others may point to the physical laws which describe the behavior of mechanical systems (that is: Newton's laws), and more extensive descriptions of creating conceptual models from real systems using these laws.

		MECHANICAL SYSTEM EXAMPLES		
		1st-Order	2nd-Order (undamped)	2nd-Order (damped)
INPUTS	0	Book Sliding on Table		
	$\delta(t)$			Pendulum
	$u(t)$			
	$u(t)\cos(\omega t)$			
	$\cos(\omega t)$			Rotary-Mechanical System

Figure Eight: Examples Matrix Page for Mechanical Systems

On the Examples Matrix Page for mechanical systems, any number of systems may be represented. Given inputs from authors, the matrix can be expanded to handle: higher order systems; other specific mechanical systems examples; videos or simulations of the dynamic behavior of these systems; and responses of these systems to inputs other than those shown.

Once the student chooses the Pendulum example, and has created a conceptual model for the real pendulum based on the physical laws of mechanical systems, it is time to extract the mathematical model, or differential equation, which describes the system's dynamic behavior, and solve for the predicted response of the system. Figure Nine describes this process in detail. Each step, again, becomes a point of departure. There is also opportunity for more descriptive comparison and contrast of the different methods, to achieve a higher degree of sophistication, and assist students in choosing the most appropriate methods for a given set of boundary conditions, for a particular system, or for a specific input forcing function.

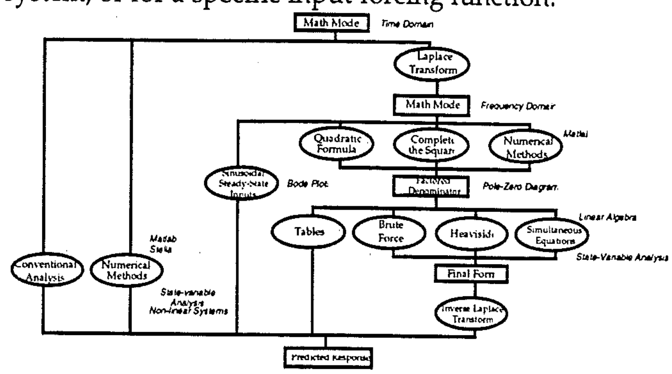


Figure Nine: Mathematical Models and Process Page

The student may choose to remain in the time domain in order to effect a solution of the differential equation. In this case, the student moves to the Time Domain Representation Matrix page (Figure Ten), and can explore the various possible methodologies. Or, the student may choose to utilize Laplace Transform methods, in which case Figure Eleven is the appropriate next step. In both cases, these time and frequency matrices serve as points of departure for finishing off the solution of the problem, and finding the predicted response. Once the predicted response has been achieved -- perhaps by using Matlab or another numerical solution tool -- it can be compared to the data taken previously. If satisfactory agreement is not reached, the student can return to the top level of the process, or any intermediate level, at any time.

		TIME DOMAIN SYSTEM REPRESENTATIONS		
		1st-Order	2nd-Order (undamped)	2nd-Order (damped)
INPUTS	0	$\frac{dx}{dt} + ax = 0$	$\frac{d^2x}{dt^2} + ax = 0$	$\frac{d^2x}{d^2} + b\frac{dx}{dt} + ax = 0$
	$\delta(t)$	$\frac{dx}{dt} + ax = \delta(t)$	$\frac{d^2x}{dt^2} + ax = \delta(t)$	$\frac{d^2x}{dt^2} + b\frac{dx}{dt} + ax = \delta(t)$
	$u(t)$	$\frac{dx}{dt} + ax = u(t)$	$\frac{d^2x}{dt^2} + ax = u(t)$	$\frac{d^2x}{dt^2} + b\frac{dx}{dt} + ax = u(t)$
	$u(t)\cos(\omega t)$	$\frac{dx}{dt} + ax = u(t)\cos(\omega t)$	$\frac{d^2x}{dt^2} + ax = u(t)\cos(\omega t)$	$\frac{d^2x}{dt^2} + b\frac{dx}{dt} + ax = u(t)\cos(\omega t)$
	$\cos(\omega t)$	$\frac{dx}{dt} + ax = u(t)\cos(\omega t)$ *Ignore Boundary Conditions*	$\frac{d^2x}{dt^2} + ax = \cos(\omega t)$	$\frac{d^2x}{dt^2} + b\frac{dx}{dt} + ax = \cos(\omega t)$

Figure Ten: Time Domain Representation Matrix Page

		FREQUENCY DOMAIN SYSTEM REPRESENTATIONS		
		1st-Order	2nd-Order (undamped)	2nd-Order (damped)
INPUTS	0	$X(s) = \frac{x(0)}{s + a}$	$X(s) = \frac{s\,x(0) + x'(0)}{s^2 + a}$	$X(s) = \frac{(s+b)\,x(0) + x'(0)}{s^2 + bs + a}$
	$\delta(t)$	$X(s) = \frac{1}{s + a} + \frac{x(0)}{s + a}$	$X(s) = \frac{1}{s^2 + a} + \frac{s\,x(0) + x'(0)}{s^2 + a}$	$X(s) = \frac{1}{s^2 + bs + a} + \frac{(s+b)\,x(0) + x(0)}{s^2 + bs + a}$
	$u(t)$	$X(s) = \frac{1}{s(s + a)} + \frac{x(0)}{s + a}$	$X(s) = \frac{1}{s(s^2 + a)} + \frac{s\,x(0) + x'(0)}{s^2 + a}$	$X(s) = \frac{1}{s(s^2 + bs + a)} + \frac{(s+b)\,x(0) + x(0)}{s^2 + bs + a}$
	$u(t)\cos(\omega t)$			
	$\cos(\omega t)$	$Y(s) = \mathcal{L}\{\cos(\omega t)\}$ $X(s) = G(s)\,Y(s) = \frac{1}{s + a}Y(s)$	$X(s) = \frac{1}{s^2 + a}Y(s)$ $x(t) = \lvert G(j\omega)\rvert \cos(\omega t + \angle G(j\omega))$	$X(s) = \frac{1}{s^2 + bs + a}Y(s)$

Figure Eleven: Frequency Domain Representation Matrix Page

Clearly, the opportunities for additional contributions to this framework are enormous. Video, sound, still photos, experiments, numerical examples, worked homework problems and exam problems, open-ended analysis or design problems -- all may be incorporated, with full hyperlinks. Such links, of course, may point outside this Systems database, as required or desirable. Historical examples and anecdotes can be incorporated.

This framework for studies of engineering systems focuses on the *process* of solving and analyzing systems, based on the foundations of ordinary differential equations (ODEs). ODEs become the unifying principles for studying systems from every discipline. As a result, the framework is rich and complex, yet has a common and unifying point of departure for the study of all systems. The framework is flexible, in that other systems, other analytical techniques (e.g. numerical methods), other examples, other perspectives (e.g. historical) can all be added with relative ease, by virtually any user or prospective author.

V. Conclusions

We have presented our concept for the authoring and retrieval of textual information which, in the past and even present, would be presented as a bound textbook. Given the power of present and future technology, however, the restrictions of this format have truly become antiquated and obsolete. We envision replacing the traditional means of authoring and publishing, by a new set of frameworks. These frameworks will create dynamic, living databases of information, textual in nature, which address the needs and sophisticated expectations of a large, computer-literate audience.

We have begun the process of developing one such database, using the body of knowledge termed Engineering Systems Analysis and Design as our point of departure from tradition.

Finally, we note that American institutions of higher learning will be under increasing financial pressures in the next decade. The problems faced by government and industry over the last several years cannot be avoided by academe. We believe our concept can facilitate this downsizing trend by decreasing the cost of access to textual information for students and researchers, and by extending the useful life of information rendered into textual form. At the same time, it will maintain high publication standards, and incorporate new and stimulating technology. Productivity will be increased by consolidating repetitive and redundant commercial resources and distributing tasks, such as document preparation and proofreading, to authors, publishers, and users of the textual information.

Acknowledgments

Numerous discussions with colleagues across the country have refined our ideas, first sketched out during the American Society of Engineering Education conference in Edmonton, Alberta in June of 1994. The feedback of John Erickson and Dan O'Connor of the Interactive Media Lab at Dartmouth deserves special recognition.

References

[Hen94] Keith Henderson, "Turning Failure into Physics Success". *The Christian Science Monitor.* Monday, December 12, 1994, p. 12.

[Maz91a] E. Mazur, "A hypermedia approach towards teaching physics". In: *Dig. Symp. Antennas and*

Propagation Soc. (IEEE Press, New York, 1991), p. 261 (vol.1).

[Maz91b] E. Mazur, "Can we teach computers to teach?" *Computers in Physics* **5**, pp. 31-8 (1991).

[Red93] E. F. Redish and J.M. Wilson, "Student programming in the introductory physics course: M.U.P.P.E.T". *Amer. J. Phys.* **61**, pp. 222-32 (1993).

[Red94] E. F. Redish, "Implications of Cognitive Studies for Teaching Physics". *Amer. J. Phys.* **62**, pp. 796-803 (1994).

[Moo92] L. Moore and D. Smith, "Project CALC: calculus as a laboratory course". In: *Proc. 4th Int'l. Conf. Computer Assisted Learning* (Springer-Verlag, Berlin, Germany, 1992), pp. 16-20.

[Har95] Kathleen M. Harmeyer, Marion O. Hagler, William M. Marcy, and Kathryn Wetzel, "Multimedia Development for Engineering Faculty". Workshop funded by the National Science Foundation, July 24-28, 1995. Joint project between ExperTech Inc., Texas Tech University, and Amarillo College.

[Lar94] Timothy R. Larson, "Making an interactive calculus textbook". In *Proc. Interactive Multimedia '94* (Society for Applied Learning Technology, Warrenton, VA, 1994), pp. 56-59.

[Ami94] Abbas Aminmansour, "Development of an interactive multimedia book in engineering". In *Proc. Interactive Multimedia '94* (Society for Applied Learning Technology, Warrenton, VA, 1994), pp. 70-73.

[Eri94] John Erickson, Ph.D. dissertation proposal, Dartmouth College, 1994.

[Eri95] John Erickson, personal communication.

[GNA94] C. Butts, C. Reilly, M. Speh and J. Wang, "WWW and the Global Network Academy". In: *Proc. 1st WWW Conf.*, 25-27 May 1994, Geneva, Switzerland.

[Mit94] Kevin A. Mitchell, "GIFConverter 2.3.7". Copyright 1988-93 by Kevin A. Mitchell (74017.2573@compuserve.com).

[Kal94] Kaleida, Inc., "ScriptX". Copyright 1995 by Kaleida, Inc.

[SGI95] Silicon Graphics, Inc., "WebFORCE". Copyright 1995 by Silicon Graphics, Inc.

[O'Co95] Daniel C. O'Connor, Ph.D. dissertation proposal, Dartmouth College, 1995.

[Kel94] Kevin Kelly, *Out of Control: The Rise of Neo-Biological Civilization* (Addison-Wesley, Reading, MA: 1994)

[Mne94] Mnemotrix, Inc., "Texis". Copyright 1995 by Mnemotrix, Inc. Texis incorporates the search engine Metamorph.

Report on European Projects in Electronic Publishing

Dr. Nikitas Kastis, Lambrakis Research Foundation

It is generally accepted that there is a large volume of cultural material, regarding not only the history of Europe but also the up-to-date cultural production, in Fine Arts, Music, etc. This is a considerable "body" of knowledge which constitutes the European Cultural Heritage, a worthy repository of concepts, ideas, life-styles, with an important economic aspect.

Considering the above-mentioned reality, among others, in the European Union, several Directorates of the Commission have already stated policies in the area of electronic publishing, in other words policies for the multimedia services industry. A number of Programmes, *e.g.* the "Information Technology" Programme (DG III) and the "Telematics Applications" Programme (DG XIII), as well as special Initiatives in European and national level, are running or are under planning in order to finance projects particularly oriented to the development (publishing) and evaluation of cultural, educational and entertainment goods, using what we call "new technologies" (IT&T).

The most well-known European projects (some of them funded by the Commission in the framework of RTD Programmes) that have ended with some concrete results in electronic publishing terms, are the two LaserDiscs "World of Vikings" (cooperation between the Interactive Media Unit of Denmarks Radio and the "York Archaeological Trust" from UK) and the "Ethnology of Greenland", both published in Denmark, the "Micro Gallery", a publication of Microsoft with some of the exhibits of the National Gallery in London, the CD-ROM "Anglo-Saxons", produced by Anglia Multimedia, the "NARCISSE", a CD-ROM with the most exciting paintings of the Louvre Museum, published by the French software house EURITIS, the two CD-ROMs "Le Louvre" and "Poussain", publications of RMN (publishing house of the French National Museums), the first one with images of some of the Museum's exhibits and the second with the work of the great French artist, and the CD-ROMs "SOPHIA" and "Logomathia", published in Greece, the first, in Greece. Both of these multimedia publications were designed to work as educational tools, the "SOPHIA" for learning Byzantine History (ages 14-20) and the "Logomathia" for learning the Greek Language (ages 8-14).

Apart from the above, the European status is also consisted in a number of the so-called electronic editions, mainly produced by the well-known "traditional" publishing houses, such as the Oxford University Press, the Elsevier, the Springer-Verlang and the Matra Hachette, which, up-to-now have focused their involvement in the area by developing electronic reference titles, mostly with textual material (*e.g.* the work of Goethe, vocabularies, etc.). Other publishers such as the Garimaldi, the John Wiley & Sons, the group Rizzoli Corriere della Sera, El Mundo, DeTeBerkom, Macmillan Publishers UK, Blackwell Publications, etc., have already entered the field of electronic publishing, trying to exploit synergies and funds from the RTD Framework of the European Commission, while either competing or sometimes cooperating with their major rivals, coming from the software industry, the telecommunication services industry and, of course, from the advertising and movie industry.

It seems that the European Cultural Heritage, being a valuable asset, mostly in the hands of the state administrations, in the various countries, constitutes a very crucial factor for the viability of the electronic publishing industry, in Europe. At the same time, the educational and entertainment needs, of every society world-wide, seem to increase rapidly, getting more demanding in (technology and) quality terms. Both the sides of the information market — the "supply" (electronic editions with cultural/knowledge content) and the "demand" — are not yet rationalized, but being unbalanced. That is the reason why

certain coordinating government (state) policies should be adopted, not only in Europe, where there is a long tradition of state intervention in cultural and educational issues, but also in the US (see the National Initiative for the "Humanities and Arts on the Information Highways", in the framework of NII), to ensure synergies and compatibilities in the field.

Some generic guidelines, adoptable worldwide, to support the maturity of the telematics technologies in the electronic publishing area, must be content specific and user-oriented, and could include:

- the strategic planning and the funding of pilot programmes for projects which could work as reference work as well as awareness and demand activating mechanisms, with an impact on scientific research and education,
- the support of private initiatives, *e.g.* in specific cultural areas, where already exists the proper infrastructure, to widen the market penetration of cultural goods (business objectives), and
- the introduction of educational and research material in electronic form in the country's educational system — especially in countries such as Greece where the vast majority consists of public schools — in order to increase demand.

The fore-mentioned practice has to be also enriched by other supporting measures such as the rationalization of the cultural material administration, the copyright and the IPR status Europe-wide (see the results of the RTD project "CITED — Copyright in Transmitted Electronic Documents", inform. by the British Library) as well as of the telecommunication infrastructure, the later being the most crucial production parameter in the years to come.

New technologies imply rapid changes not only in the organisation of our work but also in the way we communicate and we "transmit" knowledge, where exactly the concepts, new practices and techniques of electronic publishing apply. The "transmission" of knowledge is the field of the convergence of the two, distinct to first look, sectors of Culture and Education, the most promising market area of electronic editions.

PANELS

Electronic Journals: For Whom the Bell Tolls

Donald Kreider, Dartmouth College (Moderator) Don Albers, MAA
Ed Murphy, PWS Publishers Dave Rodgers, U. of Michigan, AMS
Herb Wilf, U. of Pennsylvania

Rapid growth in electronic dissemination of text and graphics has resulted in the founding of electronic journals in computer science, mathematics and many other fields. A panel of scholars and publishers examines how the role of authors and publishers will change in these new electronic waters. What is a publication? What are the real costs of electronic publication? How are copyright and other issues of intellectual property to be handled in the '90s?

The four panelists will speak to the following points:

1. **Don Albers** will speak to the *price of electronic conversion.* Scholarly societies that publish research journals are being urged by some to convert paper journals to electronic format. At what velocity and at what price is this conversion occurring? A summary of velocity, price, and benefits of conversion will be given.

2. **Ed Murphy** will speak on the possibilities for and benefits of the electronic publication of mathematics tutorial materials. He will also discuss other publication possibilities.

3. **Herbert Wilf** will talk about his journal, the Electronic Journal of Combinatorics, that is quite active and is well into Volume 2 after about one year on the Web. He will emphasize operating experiences, rather than global "is it possible?" types of questions. The journal has gained large readership and has a number of interesting and unique features. Some of these, as well as some of the unique problems that had to be faced, will be discussed.

4. **Dave Rodgers** will consider how traditional roles of publishers are being reexamined and questioned. Electronic media make some traditional services provided by publishers unattractive or uneconomic. Scholars and librarians have signaled their intent to behave as an emerging market, ready to partner with technologists, and ready to renegotiate their traditional relationships with publishers. At the same time, new interests and new players all around look to exploit these opportunities. Publishers will need to reanalyze their role and services, and adapt them to changing needs and the evolving role of information in society. They will need to hone their skills for creative pricing, packaging, and managing risk in a world where the barriers to electronic access and self-publishing are likely to fall.

Scholarly Electronic Publishing and Access: New Models from Publishers and Librarians

John R. James, Chair* Janet Fisher† Carol Magenau‡
Keith L. Seitter§

Abstract

Scholarly electronic journals are emerging as a new part of the publishing world and the virtual library collection. Models and ideas for handling electronic journals are being developed, and several of these will be presented. However, there are many unresolved issues and problems that are of concern to both publishers and librarians. The panelists will present their views on a variety of these issues, including production, quality, access, cataloging, copyright, and archiving of scholarly electronic journals.

John R. James, Panel Chair

JOHN R. JAMES is Director of Collection Services, Dartmouth College Libraries. Before coming to Dartmouth in 1983, he was Head of the Serials Division, University of Washington Libraries, and Head of the Serials Division at the University of Arizona. He has an MA in Linguistics and an MA in Library Science. John James has been active in committees of the Research Libraries Group and the American Library Association, and has written on the topic of collection management of serials and journals.

*Dartmouth College
†The MIT Press
‡Dartmouth College
§American Meteorological Society

Panelists:

Janet Fisher
MIT Press and Electronic Publishing

JANET FISHER is Associate Director for Journals Publishing at MIT Press. She was the Journals Manager and Journals Production Manager at the University of Texas Press before coming to MIT Press. Abstract: Janet Fisher is very involved with MIT Press's electronic journals projects, including the Chicago Journal of Theoretical Computer Science. She will discuss the MIT Press electonic journals projects and retooling for electronic publishing.

Carol Magenau
Acquisition and Cataloging of Electronic Journals: from MARC formats to Hyperlinks

CAROL MAGENAU is Assistant Acquisitions Services Librarian at Dartmouth College, and an active member of the North American Serials Interest Group. She has held cataloging, reference, and acquisitions positions dealing with serial publications, and has also worked at the libraries of the University of Connecticut, Northeastern University, and the Harvard Divinity School. Carol holds an undergraduate degree from Harvard/Radcliffe, and masters degrees in library science and business administration from Simmons and the University of Connecticut.

Librarians have traditionally been concerned with acquiring, organizing, promoting, and preserving the artifacts of human communication. One of the chief vehicles of communication has been the printed journal. The

versity of Chicago in 1982. After serving one year as a Geophysics Scholar at the Air Force Geophysics Laboratory, he joined the faculty of the University of Lowell (now the University of Massachusetts at Lowell) as a professor of meteorology.

Scientists working in the interdisciplinary areas referred to as the earth system sciences are using a variety of sophisticated techniques to manipulate and visualize the enormous volumes of data being produced by satellites and other observational platforms and by computer simulation models. They are routinely forced to compromise the informational value of the presentation of their results when published because of the constraints of the printed page. The electronic journal being developed as a collaboration of five scientific societies (American Meteorological Society, American Geophysical Union, Association of American Geographers, Ecological Society of America, and The Oceanographic Society) will allow scientists to publish their work taking full advantage of the sophisticated graphics they routinely use in their research. The journal will be published as a World Wide Web document. It will be produced in SGML and it is anticipated that the journal will take advantage of full SGML viewers that are now becoming available. The journal is still in the final planning stages now, but should be formally launched in mid- 1995. This is not an experimental electronic journal project, but a full production scholarly journal that is intended to serve the earth system science community as a major avenue for the dissemination of research results.

rise of the electronic journal has been widely touted as a paradigm shift, because it introduces new possibilities that have the potential to radically change scholarly communication, the role of libraries, and the economics of scholarly publishing. Issues that concern librarians include: how best to facilitate the use of electronic resources; the library's role in "selecting" remote resources; maintaining current information on materials not housed or controlled locally; the assurance of textual integrity in an electronic environment; the identification of different versions of a publication, depending on the mode of access, inclusion of graphics, hypertext, etc.; providing linkages to enhance access (e.g. connecting an article with subsequent discussion about it); determining what is appropriate for long-term preservation and who will take responsibility; and not least, budgetary constraints and the changing roles of libraries, publishers, and commercial information providers. All of these constitute challenges that are under discussion in the library world. An individual library's response to some of these issues will be profiled, as well as projects in the library community that are currently addressing the control of electronic journals.

Daniel T. Richards
"Fair Use" in an Electronic Environment: an Exploratory View from the Health Sciences

DANIEL T. RICHARDS was appointed Director of the Biomedical Libraries at Dartmouth College in 1991. Prior positions include Collection Development Officer at the National Library of Medicine, Assistant Health Sciences Librarian at Columbia University, and a variety of positions at UCLA. An active member of the Medical Libraries Association and the American Library Association, he is presently serving a three year term on MLA's Board of Directors. He has been a consultant to publishers and universities and has written extensively on a wide range of topics in librarianship. He holds the MS degree in library science from the University of Wisconsin and has done postgraduate work at the University of Maryland.

Librarians play a vital role in providing both print and electronic resources, in original form and as copies, to library users in the institutions served by libraries. These copies are made under prevailing "fair use" law which permits single copies for education and research purposes. Librarians have been scrupulous in following the fair use concept with print materials. The stated purpose of copyright is to promote the public welfare through the advancement of knowledge. Copyright law was established to balance the rights of copyright owners with users of materials under copyright. This position has been supported by librarians and library associations for many years. The balance of rights is threatened in an electronic environment and a vigorous debate is under way, the results of which could mean the end of "fair use." Librarians must educate library users of their rights and obligations with regard to "fair use," as well as the consequences of silence on this matter. The preservation of the concept of "fair use" in an electronic environment is critical to the public interest.

Information access and management in the health sciences is a critical component of all health care processes and informed health care decisions are key for high-quality care and cost containment. From the perspective of the medical librarian, this paper explores some of the "fair use" debate issues including licensing and its potential impact, the retention of a market share by publishers, and the concern of authors and publishers about redistribution of information in an altered form. These concerns will be contrasted with the benefits to patient care, health professional education and medical research which are predicated on unimpeded access to biomedical information and knowledge.

Dr. Keith L. Seitter
Earth Interactions, An all-electronic, peer-reviewed, scientific journal published as a collaboration of five societies and delivered via the Internet

Dr. KEITH L. SEITTER is Associate Executive Director of the American Meteorological Society and is Director of AMS publications. He received a B.S. in meteorology from the Pennsylvania State University in 1978, and a Ph.D. in Geophysical Sciences from the Uni-

Emerging User Interfaces For The Information Superhighway

Robert Jacob, Tufts University (Chair)
Fillia Makedon, Dartmouth College
Hermann Maurer, Graz University of Technology
Sha Xin Wei, Stanford University   Timothy Lenoir, Stanford University
P. Takis Metaxas, Wellesley College

Abstract

One of the keys to the success of using the Information Superhighway is the design of an "efficient" user interface: one that is easy to use, functional and which enables the automation and fast processing of material. The following four user-based perspective interfaces will be considered: for electronic journals, for digital libraries, for educational interactive multimedia material and for 3-D graphics and research in virtual reality.

Robert Jacob

A new style of user-computer interaction is emerging, which combines media beyond video and audio, such as 3D interaction, immersive (virtual environment) displays, gesture, eye movement, and other passive forms of non-command interaction. Because this new generation of "non-WIMP" interfaces increases the bandwidth between user and computer, it offers the promise of improved access and navigation within large information-rich networks in the future. I will describe the characteristics of the emerging non-WIMP user interaction style, give examples, including my research on eye movement-based interaction, and discuss the implications of these interfaces for user interface software.

Robert Jacob is on the faculty of the Electrical Engineering and Computer Science Department at Tufts University, where his research interests are user interface software and interaction techniques. Before coming to Tufts, he was in the Human-Computer Interaction Lab at the Naval Research Laboratory. He received his Ph.D. from Johns Hopkins University, and he is member of the editorial board of *ACM Transactions on Computer-Human Interaction* and former Vice-Chair of ACM SIGCHI.

Fillia Makedon

Fillia Makedon will discuss multimedia interfaces specifically designed to address the electronic publication on the WWW of Legacy information contained in the COSMIC (Computer Software Management and Information Center) Libraries. Currently, this information can include Rich Text Format (RTF), proprietary word processor formats such as Wang, Word Perfect, and Microsoft Word, LaTeX, raw text, images, and paper documents. As new documents are generated from new projects or even regular operations in a large government organization, the problem of efficient document management and fast access, as it now exists, is fast becoming intractable. A sample of desirable facilities for the user of COSMIC Libraries might be:

- visual navigational aids for access to information on demand.
- image and text retrieval mechanisms to access specific stored information based on user queries.
- online visual browsing tools to enable the use of COSMIC Libraries as a mechanism for teaching and research.
- new indexing technologies, provide automatic indexing, tagging and linking tools, to enable the formation of "balanced"

clusters of information that continue to reside in their original location.

- strategies for easy document maintenance.
- strategies and tools for sharing information (doing collaborative work) with an array of document developers.

The discussion will consider issues related primarily with information which is stored in the form of text or images COSMIC (Computer Software Management and Information Center) Libraries. It will review ways that would bring these libraries up to date with current technology, define intelligent dissemination mechanisms and identify research areas necessary for the wide document access using World Wide Web technology.

Hermann Maurer

A number of attempts to publish books and journals via the Internet are currently being carried out. It is our claim that much of the euphoria concerning such efforts is premature. Internet publishing today resembles what happened after the invention of the printing press: everyone published leaflets and pamphlets, without quality assurance and coordination. It took some time for book publishers and newspapers to appear as focal points assembling and pre-selecting material suitable for various audiences. Internet is much in the same situation today: potent groups that bundle and select material for different tastes and needs are still mostly lacking.

We will also mention the spectrum of different approaches to publishing journals via the Internet ,and why we believe approaches such as JUCS[1]) are more likley to succeed than others. An important aspect of electronic publishing is billing for such services. We contrast the doubtfull idea of charging for each file accessed to the alternatives "subscription for n simultaneous users" and "individual subscription": the first has been used successfully for a number of (German) reference books, the second is used e.g. for the proceedings of ED-MEDIA'95, the world conference on educational multi- and hypermedia (June 17-21, Graz, Austria): the conference volume is available on the net for all those who have registered for the conference under the URL's mentioned above for JUCS with `CEdmedia` replacing `CJucs_root`. (General information is also available for everyone[2]). We finally mention some functionalities required in digital libraries that are often overlooked, particularly the "domain specific active background".

[1] `http://hyperg.iicm.tu-graz.ac.at/CJucs_root` or `http://dragon.acadiau.ca:8000/CJucs_root`

Timothy Lenoir and Sha Xin Wei

This talk presents MediaWeaver – a framework for composing distributed media, in the context of the SiliconBase project. SiliconBase is a research project in the history of Silicon Valley, conducted by members of the Program in the History and Philosophy of Science. The MediaWeaver mediates between network services, commercial software, and interface kits with which multimedia authors and designers may easily fashion radically different interactive views into shared mediabases. The network services include search engine abstractions, filters, and relational modeling frameworks. Faculty and student authors compose distributed media using Macintosh, NeXTSTEP and World Wide Web applications, supported by services from common UNIX workstations.

The MediaWeaver is designed for fluid interactive spaces which may include, as special cases, traditional hyperdocument structures. MediaWeaver forms part of a flexible infrastructure for networked scholarly workspaces which can accomodate novel ways of interacting and communicating, changing technologies, and yet guarantee the survival and dissemination of intellectual content. Other projects supported by the MediaWeaver include a Chicana Art project, Elizabethan Renaissance Theater, experimental electroacoustic music, and an Information Map Project for Conservation Studies in South and Central America. With these projects, we are exploring how notions of writing, authorship, publication, simulation, research and teaching are evolving in these fluid media.

[2] `http://hyperg.iicm.tu-graz.ac.at/CEdmedia`

Timothy Lenoir Background

Timothy Lenoir is Professor of History and Co-Chair of the Program in the History and Philosophy of Science at Stanford University. He is the author of *The Strategy of Life: Teleology and Mechanics in Nineteenth Century German Biology*, Dordrecht and Boston: D. Reidel, 1982 (paperback edition by the University of Chicago Press, 1989), which examines the development of non-Darwinian theories of evolution, particularly in the German context during the nineteenth century. His other books include: *Politik im Tempel der Wissenschaft: Forschung und Machtausubung im deutschen Kaiserreich*, Frankfurt/Main: Campus Verlag, 1992; *Instituting Science*, Stanford: Stanford University Press, 1995 (in press), a volume which examines issues related to the formation of disciplines and the role of public institutions in the construction of scientific knowledge; and *Reforming Vision: Optics, Aesthetics, and Ideology in Germany 1845-1890*, to be completed in 1995. Lenoir is currently engaged in an investigation of the introduction of computers into biological research and the development of computer graphics and imaging devices in the biomedical sciences from the early 1960s through the 1980s. His most recent paper is on the development of nuclear magnetic resonance as a tool for chemical research at Varian Associates during the 1950s and 1960s. In conjuction with these projects, together with colleagues from Academic Software Development at Stanford Lenoir is currently constructing a multi-media research database for the history of Silicon Valley.

Sha Xin Wei Background

Sha Xin Wei is a researcher in Academic Systems Development at Stanford University. He is interested in differential geometry, aspects of mathematics and scientific simulations, and media theory. Recently, he has been designing and building – with colleagues in Academic Systems Development – a framework for authoring simulations using distributed media. They are constructing, with others, environments suitable for scholarly work in, for example, geometry, theater and history. To inform this work, he is studying issues related to human-computer interaction, richly structured media, interactive narrative, geometric visualization and other ocularcentric practices.

P. Takis Metaxas

P. Takis Metaxas is an assistant professor of Computer Science at Wellesley College. He studied Mathematics in the University of Athens before coming to the U.S. to study Computer Science at Brown University in 1985. He graduated with a Ph.D. in Computer Science from Dartmouth College in 1992. Metaxas is interested in Parallel Computing, Multimedia and Algorithm Visualization, and Computer Science Education. Specifically: Parallel Graph and Combinatorial Algorithms, Computing Issues of Parallel Machines, Parallel Algorithmic Techniques and Paradigms, Architecture-Specific Parallel Algorithms and Implementation, Realizable Models of Parallel and Distributed Computation, Development of Tools for Visualizing Sequential and Parallel Algorithms, CS Curriculum Development, Teaching Methods and Tools. He is a member of ACM and SIGACT's electronic publication board and of the DAGS steering committee. He is also an editor of the electronic *Journal of Universal Computer Science* (J.UCS), published by Springer-Verlag.

Invitational Publishing on the Worldwide Web

New Modes, New Strategies, New Audiences

PARTICIPANTS...

Panel Chair **Mike Palmer** directs the User Interface & Digital Media Lab at the Center for Advanced Technologies, the applied research facility and emerging technologies "think tank" of **American Management Systems, Inc.**, an international management and technology consulting firm.

Hope Greenberg, Humanities Computing Specialist at the **University of Vermont**, makes information technology and computer-mediated resources into practical real-world tools for her university colleagues. She speaks and writes frequently on the humanities and the Internet.

Bob Duffy is Managing Director of **Strategic Communications**, a corporate relations practice that he founded in 1988. The group specializes in building prestige and industry visibility for clients through strategic consulting and creative services both on and off theWeb.

Jenny Yacovissi is Director of Corporate Communications at **Communications and Systems Specialists, Inc.**, where she leads a team of consultants, writers, designers, and training specialists focusing on external relations, internal communications, research, documentation, and electronic publishing.

This group presentation discusses several diverse strains of multimedia publishing on the Internet's Worldwide Web: initiatives that are pushing well beyond the conventional boundaries of print —and even conventional electronic— publishing. They are:

- Primary-source "facsimile" scholarship in the humanities
- Corporate identity and "PR"
- Museum outreach and public education
- Grass-roots periodicals or "zines".

In the print realm, these activities are not often considered mainstream publishing endeavors at all. That's significant. The animating theme of our panel, which balances discussion with a hands-on demonstration of instructive and impressive examples from the Web, is that the Worldwide Web is rapidly evolving into the most innovative resource for mixed-media electronic publication anywhere.

This is true not just because the Web allows publishers to address specialist or coterie audiences to an unprecedented degree, but also because it can serve even broad audiences in ways radically different from traditional modes of print communication.

Each of the four invitational publishing disciplines we discuss here in some way cuts across the grain of conventional publishing practice —or flouts received wisdom with an offbeat or non-standard approach to communicating information.

In that respect our four electronic publishing *genres* embody the wealth of new opportunities and possibilities that the Worldwide Web represents for publishers in all segments of the economy (including the shadow economy of the Webzines). By examining the revolutionary activity at the fringes of electronic publishing, we hope to illuminate what may soon happen in the mainstream.

INDIVIDUAL PRESENTATIONS

Arts and humanities on the Web: Multimedia publication and primary-source scholarship

Hope A. Greenberg

Today scholars in the humanities are finding that the I-way provides an environment ideally suited to the pursuit of knowledge, research, and academic advancement. On-line exhibits of rare materials, in-depth collections of primary sources, large text *corpora*, electronic text centers, scholarly discussion groups, and self-publishing are just a few of the ways in which humanities scholars are leveraging the Internet's capabilities.

This presentation focuses on a number of unique scholarly resources published on the Worldwide Web — including two of Hope Greenberg's own creation. One is the multimedia *Ovid Project*, a Web-based resource focusing on illustrated editions —many of them the only surviving examples of a given edition of the Roman poet's work.

Hope's institution, the University of Vermont, houses a world-class collection of Ovid materials from which her multimedia *Ovid Project* will draw its content as it evolves. Today it includes (along with a textual commentary) electronically imaged engravings from rare 17th century German editions of Ovid's ***Metamorphoses*** by artist Johann Wilhelm Baur. Only a small set of Baur's 150 bookplates are online today: more will be digitized and brought to the Web as time passes. Ultimately, other rare illustrated editions from UVM's collection will be made available on this home page as well —a significant advantage, given the high quality and on-line accessibility of the images, for distant scholars.

Hope's other current Web project is an illustrated monograph —developed in conjunction with a UVM scholar specializing in the field— on Vermont barn architecture, an important index of rural American culture over the last two centuries. The vital element in these examples —and in the others Hope will discuss— is that the unique resources they contain can now be made available to scholars worldwide.

That advantage, exciting as it is, implies at least a few attendant difficulties. Hope will touch on some of these issues as she describes the process of bringing humanities resources online via Web-based electronic publishing.

Hope can be reached at the University of Vermont, Burlington, VT 05490. Phone: 802-656-1176. e-mail: "Hope.Greenberg@uvm.edu".

Enough about advertising and sales on the Web... How companies are using invitational publishing to build credibility and prestige

Michael J. Palmer

Popular media coverage to the contrary, there is a lot more activity among the businesses and other institutions that are establishing Web server infrastructures than just the heavily publicized initiatives designed to cash in through transactional commerce

While they are not drawing anywhere near the media attention as the cyber-merchants, other organizations are working just as hard at elaborating different segments of the business development continuum on the Web, and creating highly innovative approaches in the process. This discussion focuses on the efforts of many of these organizations to use the Web as an electronic publishing channel. Their subject matter: substantive (not to mention aesthetically appealing) multimedia information about themselves and their activities

Many organizations are promoting their own interests via Web publishing —a mode that succeeds only to the extent it conveys accurate and detailed information, with no sales hype and minimal *execspeak* news release babble. These new multimedia self-publishing modes provide an important service in the information economy, as resources for the growing community of professionals that depends on accurate data about the activities of other companies and organizations. The bottom line: Web servers are supplementing traditional self-published PR tools like the annual report, news releases, and brochures, and doing so with a potential for depth, detail, and accuracy that is far superior to that afforded by traditional print publications. A new genre of corporate multimedia publication is taking shape —and one, incidentally, that adopts many techniques from independent journalism.

Companies are building business advantage through the core values of the Web: communication and information discovery, both pursued in what amounts to a free market environment. There are no captive audiences on the Web. You can build a server and they will come, but they won't stay long if there's no information of value there. Mike Palmer will visit the leading examples of substantive corporate electronic publishing on the Web, and offer a few suggestions about the future of invitational publishing in the commercial and institutional sectors.

You can contact Mike at American Management Systems, Inc. 4050 Legato Rd., Fairfax, VA 22033. Phone:703-267-8000. e-mail: "michael_palmer@mail.amsinc.com".

Museums and the Web: Pioneering a new global publishing platform

Robert A. Duffy

More than a few early adopters in the Web community have focused on making scientific and fine arts resources from the museum realm widely available over the Net. These resources range from independently spawned virtual exhibitions —like Nicolas Pioch's much-praised (except by the French cultural bureaucracy) ***Le WebMuseum***— to official WWW servers created by renowned institutions like the Smithsonian and the UK's Natural History Museum.

Smaller collections (e.g., Emory University's Carlos Museum, Memphis State's Institute of Egyptian Studies, and Pittsburgh's Andy Warhol Museum) are fitting out hypermedia infrastructures that provide rich images of art works and much educational data —plus collateral information about their institutions' activities and programs. Their counterparts among the culture, science, and technology museums are doing much the same thing.

Bob Duffy surveys the museum Web-publishing landscape, focuses briefly on the leading exemplars, and discusses the significance of Web publishing from the standpoints of the institutional publisher and the diverse audiences that museums are striving to reach. There's no question that museum servers are vital outreach tools for their institutions. Take Pioch's mini-Louvre server. Now replicated at ten mirror sites worldwide, ***Le WebMuseum*** is likely to top a million visitors in its first year on the Net. Would any institution venture to publish an exhibition catalog —or even an illustrated brochure— in a press run that high?

This angle on the topic leads inevitably to questions of publication strategy and segmentation, content, audience size/composition, cost effectiveness, intellectual property rights, and institutional self-promotion. This latter element itself has several dimensions: public relations and image-building, fundraising, and the potential for corporate alliances and cooperative initiatives, among other isues.

Many of the leading museums, large and small, are well along in building impressive WWW access infrastructures. In the process they are participating in the creation of a revolutionary platform for multimedia-enabled electronic publishing. By the same token, their online creations offer a revealing window on the future of Web publishing in general and on the future of corporate WWW publishing in particular.

Contact Bob at Strategic Communications, 9200 Old Annapolis Road, Columbia, MD 21045. Phone: 310-596-2169. E-mail" "BOBDUFFY@interramp.com".

The WebZine Scene: Underground in Cyberspace *...or...* Eye-to-eye with Tina Brown's evil twin

Jennifer Bort Yacovissi

The Net is a powerful force for creating and galvanizing communities of interest. E-text "zines" and their emergent multimedia descendants —the Webzines— illustrate the trend well, particularly as it relates to members of the heavily hyped *Generation X.*

Underground and counter-culture publications have been around in print for years, from the days of campus radicals in the sixties. With the advent of desktop publishing the movement experienced a collective technology rush, as self-publishers gained a new set of tools for reaching their coterie audiences. Today the Net and the Web have added yet another level of empowerment, and suddenly self-publishers with the proper equipment can reach out to audiences all over the world.

As Jenny Yacovissi demonstrates in this presentation, there is a noteworthy irony here: global range doesn't necessarily mean big readership numbers. But WebZine impresarios generally couldn't care less. WebZine publishing —even more than Usenet posting— may be the nineties correlative of Warhol's fifteen minutes, or even Teddy Roosevelt's bully pulpit.

On a global network that's slowly embracing commercial elements, the Webzines are decidedly —and often stridently— non-commercial. With a few prominent exceptions, WebZines are transitory monuments to mutability itself. More often than not, they're published irregularly. They're always irreverent, almost always self-indulgent, and often maddeningly self-referential. And sometimes they are flat-out bad. But just as often they're exciting and innovative. Despite (or perhaps because of) the high "attitude"quotient of this *genre* as a whole, electronic publishers can learn a good deal from what the Webzines are up to today.

Jenny's discussion ranges widely through the Webzine landscape, and homes in a few of the multimedia gems that glisten there. Just as importantly, she comments in some detail on what these publishing endeavors, good or bad, tell us about the process of Web-empowered invitational publication in general. Among other elements, she will discuss production values, audience consciousness and circulation considerations, and the interplay between textual and multimedia elements.

Jenny can be reached at CSSI, Inc., 10260 Old Columbia Road, Columbia, MD 21046. Phone: 410-290-9500. E-mail: "jyacov@corp.cssi.net".

Expanding Museum-Based Education

Charles Fenton *(Chair)*
Cynthia Char
Bryant Patten
Jerry Romelczyk

Introduction

Acting as the traditional repositories for information and content, museums and libraries face a crisis in the information age. Heritage interpretation institutions have concentrated their collections and exhibits upon the real objects of history and science while libraries organized their holdings based upon paper printed texts. As their facilities grew, mass storage became the dominant use of space. In many facilities, access was limited to researchers as the fragile nature of the stored objects limited their availability to the general public. This use pattern was contrary to the openness of information developing in the society at large.

What role will these institutions play in the world dominated and linked by the Information Superhighway? The current merger of computer multimedia and wideband telecommunications is giving our society the impetus to acquire experience through electronic simulations of reality. In light of these developments, this panel will discuss the potential impact of information technology upon libraries and museums and offer some model systems and cost effective programs which smaller institutions can use to begin the adaptation process. The panelists will bring a unique blending of experience and viewpoints on the evolving role of museums and libraries in our culture.

Panelists

Bryant Patten is an experienced software developer and a graduate of the Thayer School of Engineering at Dartmouth College. Bryant's experience with computer multimedia began in 1983 and includes publication of his first product, *Point of View,* which was nominated for Educational Product of the Year in 1990. In 1991 he began the operation of his current business, Bryten, Inc., a company formed to find multimedia solutions for publishing, education and business clients. The company has found an international client base and has developed successful products for D.C. Heath, Oxford Analytica, Apple Computer and Sun Microsystems. The Bryten team is currently developing a next generation authoring tool for multimedia development.

Cynthia Char, an independent education consultant, is currently a Senior Associate with the Education Development Center in Newton, MA. She has been involved in the development of educational software and multimedia for children and teachers for two decades. Cynthia holds a graduate degree in Human Development from the Harvard Graduate School of Education and has achieved a highly respected position in the world of education development. She has worked as a Researcher and Media Designer with Bank Street College of Education in New York City, producing digital video story composition tools, video disks in science and art, and directed development of "The Voyage of the Mimi, and elementary science curriculum based upon a PBS series and an innovative software model. Cynthia has worked with the Children's Television Workshop, the Center for Research on Children and Television and is a member of the review board for *Journal of Computing in Childhood Education.*

Jerry Romelczyk, Library Director at the Walpole Public Library in Walpole, MA., has built the community network called WINET which serves the greater Walpole area. Jerry has formed a development coalition in his community consisting of the Walpole Public Library, the Walpole Historical Museum and the Walpole Education Department. The goal of this consortium is the development and implementation of an authoring tool for use by students and community members which will provide a mechanism for development of a history of the community. Computer multimedia has been chosen as the development platform as traditional paper and object based models have not been cost effective in assembling and authoring materials about the history of the community. The community institutions will use their traditional strengths and resources to provide the resources necessary to build the project. Jerry continues to provide the leadership necessary to bring the students, historians, users and administrators together in this innovative project.

Charles Fenton has worked with New England museums and libraries for over 20 years, first as a conservator and as Director of the Woodstock Art Conservation Center, and then as Director of Renaissance Digital, a multimedia design and marketing company, developing new resources based in education outreach. In his experiences with archives, Charles has seen first hand the dilemma which many museums and libraries face with expanding physical collections and declining revenue bases. In his recent work, he is exploring the use of World Wide Web service and computer multimedia as pathways to information traditionally bound to physical objects in the form of documents, manuscripts and three dimensional objects.

Summary

The convergence of telecommunications and multimedia will have a profound effect upon how people learn and what they learn. Museums and libraries must participate in these technologies if they are to continue to have an impact upon learning. The members will open this panel by presenting model programs which they have used or implemented with selected museums and libraries. Attendees will be encouraged to respond with this own experiences. The panel will then present examples of conceptual models providing new platforms for education and outreach. The panel will provide useful models of projects which even small institutions can implement of limited budgets.

This panel will be useful for museum curators, educators and librarians wishing to enter the Information Superhighway or looking to move in to a faster lane by implementing next generation projects.

Electronic "Texts" for Engineering Education and Technical Training: Issues and Progress

John Erickson
Graduate Research Assistant
Interactive Media Lab
Dartmouth Medical School
Hanover, NH 03755
john.erickson@dartmouth.edu
http://picard.dartmouth.edu/~oly/oly.html

Albert K. Henning
Associate Professor
Thayer School of Engineering
Dartmouth College
Hanover, NH 03755-8000
al.henning@dartmouth.edu
http://hypatia.dartmouth.edu/henning/henning.html

Mimi Jett
Chief Executive Officer
Electronic Technical Publishing
2906 N.E. Glisan Street
Portland, OR 97232
mimi@teleport.com

Robert Lynch
(formerly with McGraw-Hill)
Del Mar Publishers
Albany, NY

Thomas P. Rich
Dean
College of Engineering
Bucknell University
Lewisburg, PA 17837
rich@bucknell.edu

Engineering education at the university level, and corporate technical training in areas such as VLSI design, have a number of special needs with respect to the prospects for electronic publishing. As traditional textbook publishers turn to digital production, storage, and retrieval of knowledge-based products, there are questions of access, infrastructure, and intellectual property rights, which inhibit the development of tools for engineering education, and all areas of higher education.

More classrooms are equipped with the latest technology than ever before. Unfortunately, publishers do not have courseware available to utilize the power of these new learning environments. Some universities are developing network systems and electronic texts to deliver interactive, multimedia instructional tools to their students and professors. Some publishers are developing digital libraries and archives to deliver custom publications to their marketplace. This panel will explore initiatives to create an engineering and technical education database, as a model infrastructure for electronic technical publishing. Topics to be discussed will include:

- the role of traditional publishers;
- the role of publishing subcontractors;
- the lessons of *Primus*, McGraw-Hill's attempt to create a flexible publication environment for university faculty;
- overview and update on current publishing projects in Engineering at Bucknell and Dartmouth;
- the problems for traditional players in the publications arena -- from publishers and graphic artists, to authors and bookstores -- posed by the new publishing technologies;
- the problems posed by intellectual property considerations, and how they may be overcome;
- ideal environments for publishing, authoring, and viewing;
- and, new economies for all parties which can be attained using the new technologies satellite to the Internet.

John Erickson is a Graduate Research Assistant at Dartmouth College. His Ph.D. dissertation is focused on solutions to the copyright problem inherent in distributed curricular materials. John was an electrical engineer at Digital Equipment Corporation for eight years, prior to returning to academe. He has special interests in the use of computer technology for delivery of interactive media applications at the K-12 level. John, his wife, and two daughters live in Norwich, VT.

Al Henning is Associate Professor of Engineering at Dartmouth College. Prior to receiving his doctoral degree from Stanford University, he was a device physicist at Intel Corporation. Al received a recent NSF grant to develop an undergraduate course and workshop on micro-machining technology. He has special interests in developing new textual information vehicles for undergraduate and graduate education. Al lives in Norwich, VT with his wife, Carol Muller, and two children.

Mimi Jett is Chief Executive Officer of Electronic Technical Publishing, and of ETP/Harrison, companies that support a broad range of publishers and universities, with digital production, composition, illustration, editorial, and project management services. In addition, Mimi chairs the Small Business Forum, and current serves as an Oregon delegate to the White House Conference on Small Business. Mimi lives in Portland, OR with her husband Michael and two daughters.

Bob Lynch was, until recently, Vice President of McGraw-Hill, Inc. in charge of electronic publishing and the *Primus* project. He is now with

Del Mar Publishers in Albany, NY. He has a Ph.D. in Musicology from New York University.

Thomas P. Rich is Dean of Engineering and Professor of Mechanical Engineering at Bucknell. He earned a B.S. degree in M.E. from Carnegie Mellon University and his M.S. and Ph.D. degrees in M.E. from Lehigh University. He has held previous academic positions at Texas A&M University and the University of Southampton in England. He also worked for a number of years as a research mechanical engineer at the Army Materials and Mechanics Research Center in Boston, Massachusetts. He is currently working with Bucknell faculty to develop a set of engineering electronic texts.

Directions in Humanities Publications

Gregory Crane, Perseus Project, Tufts University (Chair)
Michael Roy, W. E. B. DuBois Institute, Harvard University
Neel Smith, College of the Holy Cross
Maria Daniels, Perseus Project, Tufts University

Gregory Crane

Gregory Crane is an Assistant Professor of Classics at Tufts University. He has been balancing traditional scholarship with electronic tools ever since programmers asked in 1982 for a graduate student to help "advise" them in modifying existing software to serve the needs of scholars. He is currently the Editor in Chief of the Perseus Project, which is about to produce a second version of its database (*Perseus 2.0*: Yale University Press). His current technological work includes the development of tools for the study of ancient science and the conversion of the standard Greek Lexicon into a database. He has also published a book on Homer, has a book on Thucydides in press and has written numerous articles on classics and on the impact of technology.

Michael Roy

One of the greatest frustrations for scholars of African-American cultural history is the paucity of first-rate bibliographic tools. This problem is compounded by the fact that up until only recently, there has also been a limited number of primary texts readily available and in-print. Much of this in recent years has been changing, and one of the great promises of new technology is that the hard work of recovering this lost heritage and publishing finding aids for the fruits of this recovery is speeded up considerably through the use of computers. Even more exciting is the possibility of representing the non-literary achievements of African-Americans through the use of multimedia technology. *The Encyclopaedia Africana* is exploring these possibilities through designing a prototype for the fufillment of W.E.B. Du Bois' dream of a multi-volume work that would catalog all that is known about the world of Africa and her Diaspora. In the course of our work, we have come up against the usual non-subject specific barriers of the technology and the law (digital video, the inability of modern copyright law to deal with digital property), while simultaneously we have enjoyed the new technologies' ability to bring to life what on the printed page is unrepresentable.

Michael Roy works at the W.E.B. Du Bois Institute for Afro-American Research at Harvard University, where he is the Director of Research for the *Black Periodical Literature Project* (directed by Henry Louis Gates, Jr. and K. Anthony Appiah), the Project Manager for *Baobab: Sources and Studies in African Visual Culture*, and Director of Electronic Publishing for *The Encyclopaedia Africana* (edited by Henry Louis Gates, Jr. and K. Anthony Appiah). He has a B.A. in Philosophy from Dartmouth College, an M.A. in English and American Literature from Duke University, and does not believe that the word edutainment should be allowed to enter the lexicon.

Neel Smith

Geographic Information Systems (or GIS) have revolutionized the study of spatially organized information. With printed maps, information cannot be separated from the map's specific visualization; with a GIS, information can be manipulated on spatially defined criteria. A printed map might illustrate archaeological sites on a topographic map, for example, but a GIS containing the same information might let you isolate sites of a particular period within a certain distance of the coast, and view those sites in a three-dimensional

perspective.

In the past, the computational demands and large amounts of storage required for many applications of GIS have limited their application to areas where the immediate cost benefits were obvious — forestry, oil prospecting, defense intelligence, for example — but as desktop machines rival the capabilities of yesterday's mainframes, GIS technology has begun to filter into unexpected areas.

Archaeologists working on the Perseus project have developed a general purpose GIS on the physical geography of the Mediterranean. In this presentation, I will focus on some of the synergies that result from having a GIS in the collection of information systems in Perseus. Examples will include using a GIS in conjunction with classical texts to explore the conceptual geography of an author; and using a cartographic front end to select material from the on-line photographic archive described in Maria Daniels's presentation.

Neel Smith is an archaeologist in the Dept. of Classics at the College of the Holy Cross. He has been a principal developer of the Perseus GIS, and his publications include articles on archaeological applications of GIS. His recent research has focused on ancient geography, especially Ptolemy and Pausanias.

Maria Daniels

The World-Wide Web provides a promising environment for the publication of color images and for the study of art history. The Web has already fostered the explosive growth of on-line art archives and art historical materials; in contrast to book publishing, electronic publishing is a medium suited to the distribution of numerous high-quality color images. The creation of larger digital libraries of pictures, particularly those of objects which are so precious or fragile as to be inaccessible to all but a small fraction of specialists, will engender major positive changes in humanities research. Digitization of manuscripts, scholars' archives, and specialized collections of objects or images, to cite a few examples, makes these materials available to a broader audience without the constraints of special-order photography, limited visiting hours, or prohibitive travel costs. Digital libraries can introduce visual materials to scholars who previously would have ignored them. Digitization also promotes the preservation of these archives by keeping physical handling at a minimum. In the near future, issues of access and copyright will be resolved so that Web users will be able to access image servers as easily as they now use on-line card catalogs or inter-library loan programs.

Over the past five years, as we have developed the Perseus Project database into an encyclopedic computer-based art history resource, we have learned a great deal about the practical aspects of electronic publication. The principal challenges facing authors of server-based art historical materials would seem to be threefold. First and foremost are concerns for the images themselves. Issues of quality, size, format, and storage need all be addressed. The ideal on-line photographic archive would contain multiple images of each object, at several levels of resolution, in order to fully exploit the electronic medium. Scholars, students, and the public should be able to get what they can't get in print: pictures of all sides or parts of an object, and the ability to look more closely at any given picture. Second, a digital library should be structured usefully, with relevant contextual information, flexible tools and resources such as unique locators, marking tools, captions, thumbnail images, and attribute or keyword search functions. Third, all these resources must be delivered in a readable, hierarchical, digestible interface with multiple display options. Art historical inquiry is based on visual comparisons, so to be useful an interface must provide the capability to locate interesting groups of pictures, place them side by side, and save selections of this data for reuse.

Maria Daniels received her B.A. in American Civilization from Brown University. She is photographer and visual collections curator for the Perseus Project, and has participated in digs in Greece and Turkey as an excavation photographer. She has created parallel film and digital archives of tens of thousands of images relating to Greek art and culture. She is currently overseeing the final stages of preparation of over 26,000 images for CD-ROM and videodisk publication as part of *Perseus 2.0: Interactive Sources and Studies on Ancient Greece*, and she is also translating these materials into a World-Wide Web image server.

The Use of Animation and Visualization in Educational Electronic Publishing

Viera Proulx, Chair[*] Harriet Fell[†] Peter Gloor[‡] Richard Rasala[§]
Marian Williams[¶]

Abstract

Visualization, Animation, and Apprenticeship in Teaching Computer Science
The goal of this panel is to show how multimedia based publications can support a new style of teaching and learning. This style is based on interactive animation and visualization, active participation of the learner, and a support for learning by aprenticeship.

Viera Proulx, panel chair

Associate Professor, College of Computer Science, Northeastern University, active in the design of interactive graphics based tools and laboratories for introductory computer science, member of ACM Pre-College Committee.

Harriet Fell

Professor, College of Computer Science, Northeastern University, active in the design of interactive graphics based tools and laboratories for introductory computer science, designer of Baby Babble Blanket - a communications tool for severely handicapped infants, with additional research interests and results in cryptography.

Peter Gloor

Assistant Vice President, Section Leader Software Engineering, Union Bank of Switzerland, one of co-authors of *Animated Algorithms* - an interactive companion to the book *Algorithms* by Cormen, Leiserson and Rivest.

Richard Rasala

Professor and Associate Dean for Undergraduate Studies, College of Computer Science, Northeastern University, active in the design of interactive graphics based tools and laboratories for introductory computer science, leader of Northeastern University Network Initiative - a plan to bring coomputer networks to all buildings, offices, and dormitories.

Marian Williams

Assistant Professor of Computer Science University of Massachusetts, Lowell and a faculty affiliate at the university's Center for Productivity Enhancement - research interests in vi-

[*]Northeastern University
[†]Northeastern University
[‡]Union Bank of Switzerland
[§]Northeastern University
[¶]University of Massachusetts, Lowell

sual programming applications, participatory design of educational software, and the graphical and auditory display of scientific information, chair of the tutorial program for the CHI '96.

Panel Summary:

This panel presentation will show several ways in which a computer can be used as a modeling and demonstration tool for the different types of dynamic processes studied by computer scientists. It will talk about the ways in which a support for interactive experimentation can enhance the value of electronic textbook beyond the level of a very fancy browsing tool.

Panel Statement by P. Gloor

Teaching Algorithms by Animation

While special purpose algorithm animation and visualization systems are available, they are still (too) little used in the classroom. Although algorithm instruction can profit greatly already today by using animations for visualizing complex concepts, there are many new application domains and technological concepts to explore until algorithm animation has reached its limits. Technical areas for research include:

- Make authoring of new animations easier: Systems as MacroMedia Director and the like allow the (relatively) painless creation of general purpose animations, but they are poorly suited to the task of animating algorithms. With special purpose tools as Balsa, Zeus, XTango on the other hand,it is relatively labour-intensive to produce new algorithm animations. We would like to see special-purpose animation tools exhibiting the ease-of-use, but exceeding the capabilities of digital movie editing systems. Going even further, we are looking for systems to build interactive exercises and quizzes, where the student can modify data structures interactively applying different algorithms.

- Extend algorithmic applications So far, we have only visualized obvious parameters like performance comparisons of the same operations on different input data sets. In the future, we would like to have a generally usable analysis visualization framework that allows for easy comparisons of, e.g., the runtime behavior of different data structures.

Panel Statement by M. Williams

Visual Programming Labs for Teaching Computer Science Concepts

The Visual Labs project seeks to help computer science undergraduates bridge the conceptual gap between learning about a concept in lecture and representing that concept in a program. Each visual lab allows the undergrads to build a graphical model, and then test the model to see whether it is correctly built. Suites of visual labs have been developed for building and testing models of computer architecture, finite state machines, Petri nets, database management systems, reactive robots, and operating systems. Empirical data show that performing the labs helps to cement the undergrads' understanding of the concepts. The research is sponsored by National Science Foundation grant number DUE-9354708, "Visual Programming Labs for Teaching Computer Science Concepts to Undergraduates."

Panel Statement by V. K. Proulx, R. Rasala, and H. J. Fell

Visualization, Animation, and Apprenticeship in Teaching Computer Science

The panelists will share their experiences in developing closed laboratory exercises for lower level CS courses with particular emphasis on using graphical presentation techniques as a pedological and motivational tool. The use of interactive animations and visualization

is combined with structured support for the student programmer to provide an apprentice style learning environment.

Three major threads form the backbone of this curriculum. The first is the use of interactive animations and experimentation programs to introduce and illustrate dynamic processes - e.g. algorithm behavior, or changes in data structures over a period of time. The second thread is the use of graphics in student programs, not only as motivation, but also as a visual feedback and debugging tool. The third thread tying all together is the extensive use of model programs, shell drivers, toolkits, and procedures that encapsulate abstractions. These programming tools support the apprentice style of learning and illustrate good software design and practice throughout the curriculum.

Students working in our labs are both users and creators of interactive animations that illustrate computer science concepts. These could be an algorithm (e.g. a minimum spanning tree), a data structure and its implementation (e.g. a heap), a programming language construct (e.g. a nested loop), or a semantic concept (e.g. passing parameters to a procedure). Other laboratories animate solutions to problems while engaging students in a creative algorithm design (e.g. The Game of Life, Swimming Fish Maze, Track Stack Sorting). Interactive animations illustrate the complexity of different algorithms designed to solve the same problem. Time Trials programs collect data about algorithms that can be displayed and analyzed by a spreadsheet program. The programs student use as well as the programs students write are all written in THINK Pascal for Macintosh computers, with QuickDraw Toolbox procedures creating the graphics.

The research has been sponsored by National Science Foundation grants number DUE-9152211, DUE-9255536, and USE-9155929.

The Freedom of the Press Project - Electronic Publishing Lessons for Libraries, Information Technology and University Presses

Southern Illinois University at Carbondale

Abstract

The Freedom of the Press Project began with a partnership between Library Affairs, Information Technology and the University Press, sponsored by the Coalition for Networked Information and the American Association of University Presses, to explore electronic publishing on the Internet. Throughout the past year, the Library has lead the project to digitize volume 1 of Ralph McCoy's Freedom of the Press, an eight thousand entry annotated bibliography on censorship and suppressed speech. A major portion of the project was to create hypertext links from the index to the annotations and further, to provide hypertext links from the annotations to the articles cited in the bibliography.

This project represents a model for developing electronic library resources by creating an environment of partnership and cooperation across departments in a university setting. By bringing together individuals from the cooperating departments into one panel, we will present an overview of this project from a variety of perspectives. We will illustrate how this project developed from a inexact idea of what might be possible to a unique scholarly resource only through cooperation and the sharing of information across departmental lines. The panel members will discuss the issues of the project that relate to their various departments.

Carolyn Snyder, Dean of Library Affairs will present the issues that concern libraries. The Internet provides opportunities for dissemination of information that have the potential to enhance access to that information. By taking an active role in developing resources on the Internet, libraries will help define the way information access is structured on the net. Libraries are no longer institutions that provide access only to physical pieces housed at a specific institution, but are now developing the concept of the virtual library. In the virtual library, patrons may access information from a variety of information providers, in a variety of forms. This new technology provides the library with the opportunity to look at the patrons needs for access to information and to explore redesign of the library structure to support those needs.

Susan Wilson, Associate Director for Strategic Planning at the University Press will discuss the changing role of university presses in scholarly publishing in the electronic age. Although publishing on the Internet provides opportunities to reach a much larger market than traditional publishing, it also introduces a new dimension for concern in copyright security. Although we intend to provide free access to the bibliography, original copyright restrictions and staff time needed to upload non-copyright protected materials will play major roles in determining how the Press can provide access to this information. If presses are to remain profitable in this electronic age, we will need to reevaluate the manner in which presses charge for their services.

Jay Starratt, Director of Technical Services and Automation for Library Affairs will address the issue of the changing role of libraries and its employees. Although information seekers will continue to walk into the library for service, they are also able to connect to the library from remote locations such as dorm rooms, offices, and homes. The library must be ready to provide reference services to meet the changing needs of its patrons. Additionally, digital technology provides opportunities for the library to provide access to unique or hard-to-access collections. In addressing the changing needs of patrons, libraries must look at restructuring work assignments across departments within the traditional library structure to more efficiently meet those needs in this new information era.

Mike Schwartz, Head of the Campus Wide Information System Team will discuss the changing role of information technology personnel in providing access to information. The partnership created between the library and information technology professionals has provided new insight into how users wish to access information. Information technology professionals now interact regularly with library professionals about not only the type of services provided, but also about how to structure information so the user can easily access it. Additionally, information technology professionals have gained valuable information about the hardware demands facing universities in providing electronic access to information.

Susan Logue, Project Director will discuss the technical aspects of the operation and will provide a demonstration of the project. As we experimented with scanning and optical character recognition software, the vastness of this project became evident. Throughout the project, the group worked with a variety of software products for Macintosh, DOS and UNIX machines to determine the most efficient method to convert printed text to electronic text. Creating a team of individuals that could work efficiently and effectively together in an

ever-changing environment presented managerial challenges.

Biographical Statements

Carolyn A. Snyder was appointed Dean of Library Affairs at Southern Illinois University in Carbondale on September 1, 1991. She has been a leader in University groups and in state and national professional organizations in planing and implementing technology-based library services.

Jay Starratt has served as Director of Technical and Automation Services in Library Affairs at Southern Illinois University since 1992. Through publication and presentation, he has focused his attention on library technology, administration and management.

Mike Schwartz has held the position of Assistant Director for the Campus Wide Information System at Southern Illinois University since 1992. Through his direction, the University has moved forward in developing an innovative campus-wide information system.

Susan Logue is the Project Director for the Freedom of the Press bibliography digitization project. She is currently the head of the Imaging Project Group in Library Affairs at Southern Illinois University.

Susan Wilson is Associate Director for Strategic Planning at the University Press. She supervises computer utilization at the Press and heads the copy editing department.

The Publishers' Perspective

Fillia Makedon, Dartmouth College (Chair)
Frederick Bowes, Cadmus Digital Solutions   Bruce Judson, Time Inc.
Brewster Kahle, WAIS Inc.   Edward Murphy, PWS Publishing
Peter Prichard, Freedom Forum

A survey of Electronic Publishing developments makes it clear that the technology alone is certainly not what will determine the success of Electronic Publishing ventures in the commercial world of publishing. Whether an electronic version of USA Today will make money is very much dependent on cultural and sociological factors which require "retraining" of readers as well as of writers. A similar situation holds for publishers of non-technical books, where a book under the arm by the beach beats any interactive multimedia interface in any office. On the other hand, there are special markets for publishers of books and newspapers, such as libraries, government institutions, networks of "converted readers" that wish to also have electronic access to books, ranging from poetry to pottery...

This panel is designed to provoke discussion on these issues and give the perspective of a diverse set of publishers. Some key issues that will be considered are:

- how paper publishing products differ from electronic counterparts
- what strategies are currently being considered as viable alternatives
- how can a publisher protect his assets and make money from an electronic publishing venture?
- How can publishers guarantee quality of product given the competition, issues of author/editor retraining and shorter marketing cycles the Internet brings about?
- How can publishers exploit new product ideas, like the individually-customized publications that electronic media make possible?

Another major issue concerns the definition of what "digital property" is and how copyright laws may need to change before they can be applied to copies of electronic materials in a commercial setting.

- Are publishers currently too obsessed with establishing property and intellect to worry about establishing quality in their Electronic publishing ventures?
- How are copyright laws to be established for digital documents and how is cost of membership to be determined?
- Are publishers equipped to cope with the new ways of content acquisition, content management, editing, production and dissemination?

These and other issues will be discussed from different publishing perspectives.

Fillia Makedon is an Associate Professor of Computer Science at Dartmouth College since 1991. Before that she was Associate and Assistant Professor at the University of Texas at Dallas and at the Illinois Institute of Technology in Chicago. She received her Ph.D. in Computer Science from Northwestern University in 1982. She is Director and Founder of the Dartmouth Institute for Advanced Graduate Studies in Parallel Computation (DAGS Institute) which was founded in 1992, jointly with Professor Donald Johnson. The institute's aim is to explore new applications and uses of high performance computing. Professor Makedon is also Director of the DEVLAB, (The Dartmouth Experimental Visualization Laboratory) which focuses on proving basic research tools and new algorithms for multimedia system applications. She is currently supervising 6 Ph.D. students and her interests are in the areas of digital video

editing, video motion analysis, information retrieval, electronic publishing and multimedia interfaces for digital library applications. She is author of numerous research articles, and recipient of many awards. She is the mother of three children, Basil, Dana and Calliope.

Frederick Bowes III, president of Cadmus Digital Solutions, is at the forefront of utilizing emerging technologies to create new publishing opportunities. He brings over 23 years in strategic and operating management in the publishing and printing industries to his new position at Cadmus, a leader in the production of scientific, technical, medical, and scholarly journals, which recently announced the creation of AtHOME@CADMUS, a new content management service on the World Wide Web.

Mr. Bowes' successful track record in implementing new technologies includes 9 years at the Massachusetts Medical Society where he served as vice president of publishing and publisher of *The New England Journal of Medicine*. While at the helm, he quadrupled *The Journal's* revenues; expanded its print publishing program with the launch of specialized association journals, several books, and a series of bound reprint collections; developed a pioneering electronic publishing program; and launched an early CD-ROM title *Compact Library: AIDS*, a comprehensive quarterly updated compendium of AIDS information and winner of the prestigious "Laserdisc of the Year" award.

While president and chief executive officer at Macmillan New Media, a developer and publisher of multimedia CD-ROM titles for the library, professional, and consumer markets, Mr. Bowes oversaw the acquisition of the medical CD-ROM business from e Massachusetts Medical Society and built it into a market leader with a profitable product line and strong international distribution; developed a pioneering, multiple award-winning children's multimedia CD-ROM, *Macmillan Dictionary for Children—Multimedia Edition*, an innovative school guidance software CD-ROM working with The College Board; as well as other CD-ROM titles for the library, professional, and consumer markets.

Active in industry associations, Mr. Bowes serves as a current delegate for the International Publisher's Association's Electronic Publishing Committee, and is a member of the Association of American Association of American Publishers (AAP) Electronic Publishing Committee. A graduate of Dartmouth College, he holds an M.B.A. from Columbia University.

Bruce D. Judson is a leading innovator in marketing and multimedia. He is General Manager, Time Inc. New Media, a division of Time Inc. Mr. Judson's responsibilities include developing Pathfinder, an Internet-based on-line service involving Time Inc. magazines. He is also active in creating interactive marketing applications for the Full Service Network (tm), Time Warner's Information Superhighway.

Mr. Judson is the author of "Effective Marketing on the Internet," a forthcoming article in *The Advertiser*. In addition, he is a frequent speaker at industry conferences on multimedia. He was recently named Vice Chairman of the Magazine Publishers of America's New Media Committee.

Earlier in his career, Mr. Judson served as Director for Marketing for Time Inc. Magazines and Director of Target Marketing for Time Inc. In these positions, he led the roll-out of selective binding and ink-jet technology to Time Inc. publications. This technology enables Time Inc. magazines to be customized for consumers and advertisers using database marketing techniques.

Mr. Judson received a law degree from Yale Law School and a management degree from Yale Management School in 1984. He was a Senior Editor of the Yale Law Journal and co-founder and Editor-in-Chief of the Yale Journal on Regulation. He is a member of the New York Bar, and a 1980 graduate of Dartmouth College. Mr. Judson also participates in charitable activities. He is a member of the Board of Directors of the National Neurofibromatosis Foundation, and was recently elected Senior Vice-Chairman.

Inventor and architect of the WAIS' electronic publishing system, **Brewster Kahle** has been an influential leader in the electronic publishing industry. He founded Wide Area Information Servers Inc. in July 1992 to create software products and consulting services to further develop the role of the WAIS in Internet publishing. As President of WAIS Inc., Brewster is pioneering new publishing

paradigms, including Internet-based information services and agent-based publishing.

Before starting WAIS Inc., Brewster was one of the founding members of Thinking Machines Corporation of Cambridge Massachusetts. He designed the chips and processor boards for the company's early supercomputers. Brewster was schooled at MIT in Computer Science and Artificial Intelligence where he worked closely with Danny Hillis and Marvin Minsky.

Edward F. Murphy is the president of PWS Publishing Company, a subsidiary of Thompson Publishing. PWS Publishers are the producers of quality educational materials for the disciplines of mathematics, engineering, and computer science.

> Educational publishers are scrambling to create a future for themselves with new media while maintaining and growing their traditional print based businesses. What are the business opportunities that make this transition worth the effort? How are publishers responding to these opportunities? Who's winning and why?

Peter S. Prichard is a senior vice president of the Freedom Forum, the world's largest foundation devoted to free press, free speech, and free spirit. At the Forum he is executive director of the Newseum, the Freedom Forum's largest operating program and the only museum in the world devoted to the past, present and future of news. The $40 million museum is scheduled to open in 1997. Before joining the Forum, he was editor of USA TODAY, the nation's largest-circulation newspaper.

> Although I believe the Internet is a wonderful resource for people around the world to exchange messages and share research, I think its potential as a consumer market has been exaggerated by the media. The Internet has a long, long way to go before it rivals the current advertiser-supported media of the world, and in fact, as a consumer product, it may be bypassed by the coming marriage of the computer, the telephone, and the television set.

Perils and Pitfalls of Electronic Conference Proceedings

Moderator: Samuel A. Rebelsky *Dartmouth College*
Panelists: Robert B. Allen *Bellcore*
Frank Baker *NCSA*
Robert Mack *IBM*
Charles Owen *Dartmouth College*

INTRODUCTION

Multiple-author works, including anthologies and conference proceedings, are forms of publishing that are particularly impacted by the advent of electronic publishing. In addition to having different ideas and writing styles, authors in an anthology work often use a wide variety of electronic document preparation systems. Even when many authors use the same system, they frequently use it in quite different ways. This increases the difficulty of the anthologists' and editors' tasks, as they must not only coordinate ideas, but also find a way to bring diverse formats together to form a coherent document.

In this panel, we will discuss electronic conference proceedings—collections of papers and related materials prepared as a record of an academic conference, most frequently for scientific conferences. Electronic conference proceedings are a particularly interesting instance of the multiple-author anthology as they are further complicated by very short deadlines and a very large number of authors.

The members of this panel will share their experiences collecting, organizing, and disseminating electronic conference proceedings from a wide variety of conferences, both large and small, including *IWANNT'93* (the International Workshop on Applications of Neural Networks to Telecommunications), *WWW2* (the 2nd World Wide Web Conference: Mosaic and the Web), *CHI'95* (the 1995 ACM conference on Human Factors in computing systems), *STOC95* (the 1995 ACM Symposium on the Theory of Computation), *FOCS95* (the 1995 IEEE Symposium on the Foundations of Computer Science), and *DAGS'93* (the 1993 Dartmouth Institute for Advanced Graduate Studies Institute on Parallel Computation and Parallel I/O). The conferences have a wide variety of audiences, including researchers in theoretical computer science, computer graphics, human-computer interaction, and hypertext. The materials have been used for a variety of purposes, including reviewing, printed proceedings, networked proceedings, and CD-ROM-based proceedings. In coordinating, collecting, editing, and disseminating conference papers, the panelists have experienced both successes and failures and hope to share their "war stories" with the audience.

The particular issues they will discuss include document formats, submission mechanisms, audience response, and implications of electronic proceedings. Questions and partial responses are summarized in the following sections.

PRESENTATION FORMAT

The presentation format for an electronic anthology must accommodate a wide variety of platforms and a wide variety of features. No document format yet supports all the expectations and needs of electronic proceedings and exists on a wide variety of platforms. At present, many conferences have chosen to use HTML (the HyperText Markup Language) because of the growing popularity of the World Wide Web. Is this the appropriate format? If not, what format (or formats) should conferences use? How much uniformity can (or should) conferences ensure?

To provide a larger feature set, the proceedings for IWANNT'93 were created in SuperBook format. This allowed the conference to provide networked proceedings that, among other things, could include shared annotations. It did, however, require software that is not used universally. However, at the time these proceedings were created, neither HTML nor the WWW had made a big impact.

To provide maximum flexibility and universal access, both the WWW2 and SC'95 proceedings are World Wide Web-compatible document sets, prepared primarily in HTML but employing other data formats as necessary. HTML offers the strength that such documents are easily accessible from most computing platforms, can easily be made available over wide-ranging networks and on CDROM, and can be created in a multitude of writing environments. HTML also offers an excellent mechanism for creating a single, structured, and cohesive document set from many otherwise unrelated documents.

However, there are disadvantages to HTML. Among its failings are insufficient support for mathematical formulae. In a scientific proceedings, such formulae are especially important, and current solutions (e.g., including formulae as inline images) are less than optimal.

SUBMISSION FORMAT

Presently, there are a wide variety of languages for creating electronic documents, including TeX, LaTeX, HTML, SGML, troff, SuperBook, and even "plain ASCII." Most authors are only comfortable with one or two such languages. While it would be wonderful to require authors to submit in whatever format is used for the final version of the proceedings, the switch to a new format may be overly burdensome to many authors. What formats should those managing submissions allow?

At present, many conferences are using HTML as a de facto standard for documents intended to be used directly in electronic proceedings. However, as many have observed, HTML is notably lacking in its ability to describe the printed page. Hence, many conferences also ask authors to submit their documents in PostScript or other more fully-featured language. The PostScript version is then used to prepare the printed proceedings, while the HTML version is used to prepare the electronic proceedings. For example, WWW2 asked for PostScript versions for the preliminary conference proceedings. STOC'95 requested PostScript for initial submissions, but returned to hardcopy for final submissions.

Some audiences are conversant in a common publishing environment and can be required to submit in a specific format. For example, virtually all members of the WWW2 audience understand HTML and all papers were submitted in that format. For DAGS95, a large portion of the community could be expected to understand and use HTML, and authors were requested, but not required, to submit HTML.

However, many audiences do not have a common background that permits a uniform submission format. For such groups, requiring a common data format creates an unnecessary obstacle to participation. For that reason, many conferences currently chose to provide substantial support to convert final papers to a uniform format (frequently HTML or another Web-compatible format). This support is made easier by the availability of conversion tools. Under such circumstances, the ability to accept papers created in a wide variety of word processing and publishing environments is critical to achieving the widest possible participation.

CONVERTING FORMATS

In the abstract, it seems simple: "let authors submit using the package of their choice and use an automatic conversion package." But how easy is it really?

For IWANNT'93, authors were invited to submit electronic versions of their papers in LaTeX, RTF, Troff, PostScript, or ASCII format. Of the 42 conference papers, electronic versions of 40 were returned. However, the authors did not always use a standard template and considerable effort was required to reformat the materials.

For DAGS'93, authors were allowed to submit their papers in a variety of formats, although most chose to use TeX or LaTeX. While automatic translation of text and figures to another format was relatively straightforward, the translation of mathematical formulae required significant effort.

For CHI'95, authors were requested to submit electronic versions in addition to printed documents. These submissions were in addition to normal printed submissions, as the conference CD-ROM will be a supplement to the traditional printed proceedings. This was the first CHI that included an electronic component and due to this novelty, our guidelines specified a limited but still diverse set of paper formats for submission. We requested, but did not insist on HTML documents. About a third of the participants submitted their papers in HTML format. The rest provided formats that allowed for relatively easy conversion. In addition, the general design policies for the HTML versions of documents had not been completely determined at the time authors were asked to submit their papers, so each paper had to be manually edited to make changes dictated by experience with the papers and more global structural issues driven by navigation requirements for this large collection of papers. A surprising number of low-level file handling and quality issues arose, requiring iteration with authors on file submission for the text and the figure and table images.

These experiences demonstrate some of the problems with relying on "automatic" conversion. They also suggest another pitfall of electronic publications: uniformity (or lack thereof). Even with work from the submissions coordinator and helpers, there is often little uniformity to electronic proceedings. Consider the case of HTML. While some authors write directly in HTML, others use an automatic conversion package, such as latex2html to build their electronic documents. Because of the variety of original formats, there is often little coherence in the final proceedings. Just as many conferences now provide "style files" for their proceedings (e.g., for the printed proceedings, CHI'95 expected, among other things, two column articles, with times text and helvetica section titles), it is likely that conferences with electronic proceedings will soon provide more extensive HTML style files.

SUBMISSION MECHANISMS

When choosing a submission mechanism, the submissions coordinator must provide for a wide variety of platforms and connectivity and ensure that some forms of security are ensured. Current mechanisms include electronic mail, ftp, and "backwards ftp" (in which the author puts the document in a publicly accessible place and the coordinator transfers it from

that site). What mechanism (or mechanisms) should be used?

For SC'95, it was decided that electronic submissions are best received via FTP, though some papers were submitted via email for both conferences. SC'95 allowed proposed papers to be submitted in hard copy, but all final papers will be submitted electronically.

For STOC'95, it was decided that electronic mail provided greater access (people who cannot ftp can still send electronic mail) and greater security (traditional ftp servers allow overwriting of files; a mail based system can avoid such overwriting with minimal effort). However, mail-based submissions did create some problems. In particular, while there is a standard for ftp, there is not one for electronic mail based submissions. While we had asked authors to register their papers electronically before submitting (so that we could create a database of submissions), many authors simply sent their papers via electronic mail, without bothering to read the instructions.

AUDIENCE RESPONSE

While electronic submission will eventually become commonplace, it is still a relatively new practice. How does the audience react to being asked or allowed to submit electronically? What problems does the audience observe? How well does the audience read directions?

Audience response to the electronic proceedings format and electronic submissions has ranged from "It's about time!" to "What a waste..." Positive responses have far outnumbered the negative. The WWW2 audience assumed that submissions would be electronic. Most members of the SC'95 audience seemed very comfortable with the electronic submission process, but many proposals were initially submitted in hard copy.

Even something as simple as electronic submissions can make a big difference. For STOC'95, over 75% of the papers were submitted electronically, even though it was the first year for electronic submissions. Although STOC'95 required a multi-step process in which authors registered their submissions before submitting, authors were extremely positive about the process (particularly since many wait until the last minute to submit, and electronic submissions therefore allowed them to save Federal Express or DHL expenses). Reviewers also seemed to respond positively, as it gave them easier and faster access to papers.

ARCHIVAL VERSIONS

For scientific documents, including conference proceedings, it is imperative that archival editions continue to exist so that scientists have a fixed version of each paper to refer to and so that they are guaranteed to have access to the materials. How do electronic documents fit into this picture?

Most conferences still treat the printed versions as the primary, archival format. A system is in place for handling such proceedings, and publishers, readers, and librarians all understand such a format. For example, the IWANNT'93 proceedings are a supplement to the main proceedings and are not intended to supplant them. (In fact, there is a question as to whether the publisher of the proceedings would have allowed an electronic proceedings if they'd felt that electronic proceedings would impact their sales.)

However, some conferences are eschewing the printed version as archival format. In particular, the archival version of the WWW2 and SC'95 proceedings are both electronic. In fact, there will be no print version of the SC'95 proceedings.

IMPLICATIONS

When authors are able and willing to submit in electronic format, how will proceedings change? Will they include papers that might not otherwise be included (as supplements)? Will they allow longer versions of papers, since there is no longer a printing cost associated with such papers?

Many conferences have just begun to "go electronic" and have not considered all the implications of electronic format. However, some have begun to explore these extended possibilities. At WWW2, members of the audience were encouraged to submit papers to include in an "extended proceedings." The networked electronic proceedings for DAGS95 will contain audio and slides from selected presentations.

Authors are also beginning to exploit the opportunities suggested by electronic proceedings. Several WWW2 and SC'95 papers take advantage of the hypermedia environment, including hyper-linked text, color images, audio, and video. Much more color is used than would be possible in an economy-minded print environment.

PANEL MEMBERS

Robert B. Allen (rba@bellcore.com) is a research scientist at Bellcore in Morristown, NJ. Bob attended Reed College and received his Ph.D. in Experimental Psychology from the University of California, San Diego.. He is Editor-in-Chief of the *ACM Transactions on Information Systems* (TOIS) and he is the General Chair of the 1995 ACM Multimedia Conference. He was responsible for the SuperBook-based networked conference proceedings for IWANNT'93 and is coordinating the development of a electronic edition of ACM TOIS.

Frank Baker (fbaker@ncsa.uiuc.edu) joined NCSA (the National Center for Supercomputing Applications,

on the campus of the University of Illinois at Urbana-Champaign) in 1993 and is a member of the newly-formed Information Technology Projects Group. Mr. Baker earned his B.S. in Math and Computer Science at the University of Illinois (1971) and a Masters of Education at the University of Massachusetts (1972). He has been a public school teacher and administrator and a technical writer in the software industry. Mr. Baker's current work is focused on hypermedia publishing and publishing in the Internet environment. He managed the creation of the electronic and printed proceedings for the Second International World Wide Web Conference -- Mosaic and the Web (WWW2) in 1994 and is currently coordinating electronic submissions for IEEE SuperComputing95 (SC'95) and the creation of the CD-ROM proceedings for that conference

Robert Mack (maier@watson.ibm.com) has been a Research Staff Member at the IBM Watson Research Center since 1981. He has a doctorate in Experimental Psychology from The University of Michigan. Robert's research interests include development of usability engineering methods, prototyping and evaluating of graphical user interface techniques, and training methods for using computers. Robert coordinated electronic submissions for CHI'95 (Association for Computing Machinery Conference on Human-Computer Interaction), and was production editor for the CD-ROM Proceedings (and Companion).

Charles Owen (cowen@cs.dartmouth.edu) is a graduate student at Dartmouth College. At Dartmouth, he is investigating multimedia information retrieval and digital video compression. Mr. Owen is also student member of the ACM SIGACT committee on electronic publishing, helped coordinate electronic submissions for the 1995 ACM Symposium on the Theory of Computation, and managed the electronic conference proceedings for the DAGS 1993 and 1994 conferences.

Samuel A. Rebelsky (samr@cs.dartmouth.edu) is a visiting assistant professor of Computer Science and assistant director of the Dartmouth Experimental Visualization Laboratory at Dartmouth College and is co-chair of the DAGS95 Conference on Electronic Publishing and the Information Superhighway. Sam received his Ph.D. in Computer Science from the University of Chicago in 1993. He is a member of the ACM SIGACT electronic publishing board, and coordinated electronic submissions for STOC'95 (the ACM symposium on the theory of computation) and FOCS95 (the IEEE symposium on the foundations of computer science) and developed the interface for the DAGS'93 electronic conference proceedings. Sam's research interests include electronic publishing, multimedia, and programming languages.

Obstacles in the Implementation of Company-wide Information Highways

Moderator

Peter A. Gloor, Union Bank of Switzerland

Section Leader, Software Engineering UBS, and Adjunct Assistant Professor, Dartmouth College. He is, among other things, responsible for introducing a company-wide WWW-based information system within UBS.

Panelists

Tim Berners-Lee, MIT

Inventor of WWW and leader of the MIT WWW Consortium.

Brewster Kahle, WAIS Corporation

Inventor of WAIS, the Wide Area Information Server, originally developed at Thinking Machines Corporation. He is currently the President of WAIS Corporation.

Jim Leavitt, Bremer Associates

Vice President of Technical Consulting and Consulting Operations at Bremer Associates, a Boston information technology consulting firm.

Abstract

This panel discusses problems related with introducing Internet/WWW-based company-wide information systems. Companies have two options: they can either decide to fully connect to the Internet, accepting thus the risks of a system that still includes security, performance, and ease-of-use flaws. The other option is to use the information highway-technology within a closed and more secure environment, thereby waiving all the advantages of being connected to the rest of the world.

Introduction

With the advent of the information superhighway, readers have at their fingertips all they need to know about electronic publishing, investment banking, or for that matter Indian culture. On-line systems like Archie, WAIS, DowQuest, etc., and most prominently World Wide Web, offer direct access to any source of information in the world. Although the Information Superhighway started out from academia, it has been embraced enthusiastically in the meantime by the business community at large. But behind the World Wide Web servers stocked with product information, marketing messages, inexpensive e-mail connections, and appealing bulletin boards lies a network that lacks auditability, privacy, reliability and security.

Auditability, privacy and security

According to the FBI computer crime unit, 80% of all reported computer crimes involve the use of the Internet to break into the target computer. Although security options, such as privacy-enhanced e-mail, exist, they are not widely deployed. To protect their resources, enterprises attaching their network to the Internet normally erect and maintain firewall gateways. But as companies already have learned, even firewalls are not impermeable.

Performance

Because the Internet is a network of networks, messages may travel through many subnetworks, some of which are operated by volunteers at universities and research centers. Although the Internet backbone is a (relatively) high-performance network, the subnetworks are not always reliable. Messages may be delayed or even disappear altogether. If video is transmitted, then even the high-performance backbone is quickly saturated. Furthermore, because of the explosive growth in Internet usage, network performance and support problems can be expected in the near future.

Ease of Use

Using proprietary enterprise e-mail and information systems is relatively easy. On the Internet, user-friendly interfaces are just emerging. To support corporate links to the Internet, it is necessary to know UNIX and Internet addressing protocols.

Another problem is known under the term "being lost in hyperspace". This means that the cognitive load of locating information can be overwhelming. Tools that assist in searching and retrieving in the information universe as, e.g., WAIS, have been available in the academic community for some time. In the business community they are just emerging.

Use Information Superhighway Internally

Once an Internet connection is in place, it can be used for a variety of applications, some of them inappropriate. For example, sensitive information may wind up being unintentionally published over the Internet. To fix this problem at least temporarily, until the general security and auditability problems will be solved, companies may use parts of the Internet technology only internally **within** the corporation.

When is it Time to Connect

Obviously if a company decides to close the gateway to the outer world, it shuts itself out of all the information available on the Internet. This can even have advantages, as employees are not wasting company time by aimlessly browsing through cyberspace. Nevertheless, in the long run a company cannot afford to close itself out of things like on-line publishing, product catalogs, order taking, customer support, database access, etc.

DEMOS and POSTERS

A Demonstration of a New System for Global Distribution of Document Images (LAROLA)

Timothy R. Thomas, Carlos I. McEvilly, Francois Laroche, Mojo B. Nichols, Jim Davies

Computer Research and Applications Group, Los Alamos National Laboratory Los Alamos, NM 87545

March 15, 1995

INTRODUCTION

The capability of delivering archival images globally over the Internet at low cost would add greatly to the usefulness of archival libraries by dramatically increasing the availability and usability of the material. The LAROLA (Los Alamos OnLine Report Archive) project accomplishes this.

MAIN RESULTS

The basic functionality of LAROLA is to provide screen viewable images of the original documents over the Internet via the WWW. In addition to this basic functionality, LAROLA also provides printable versions of the page, indexed ASCII versions, and formatted ASCII where appropriate (such as Acrobat pdf files, FrameMaker miff files, etc.).

Each page is delivered individually, thereby reducing the load on the local system, which can be as small as a normal PC or Macintosh with a modem link. Internal navigation within the documents is provided by a system of thumbnail versions of each page montaged into groups for rapid visual scanning of the document. This technique permits users of LAROLA to easily navigate within arbitrarily large documents (documents of up to 257 pages in length will be demonstrated).

Searching is done on the ASCII version (which is produced by OCR, and the information in the archival libraries card catalog) and permits full Boolean searches (AND, OR, NOT, ADJ) with wildcard capability. Fielded searches on the catalogued information (Author, Title, Year) are also provided. Searching can also be done on the text of the full document. In addition LAROLA provides fast relevance feedback searches with the user's choice of either WAIS or n-gram based document matching. Bibliographic information is a click away via automatically constructed Z39.50 queries to library catalog servers.

LAROLA is based on free software, and follows a basic philosophy of increasing usability by pre-preparation and storage of many versions of each page image. This means it must reside on a system with a very large and reliable mass storage capability, such as that at Los Alamos. However, from there it can have global distribution over redundant T-3 links to the Internet backbone. Backup and security of the archival data is accomplished as a normal course of operations of this large storage system. Access control can be configured in cases for which it is required.

Figures 1, 2, 3 & 4 show the home page, the search page, the montage page and one image preview page. However, anyone can view the system at http://www.c3.lanl.gov/larola

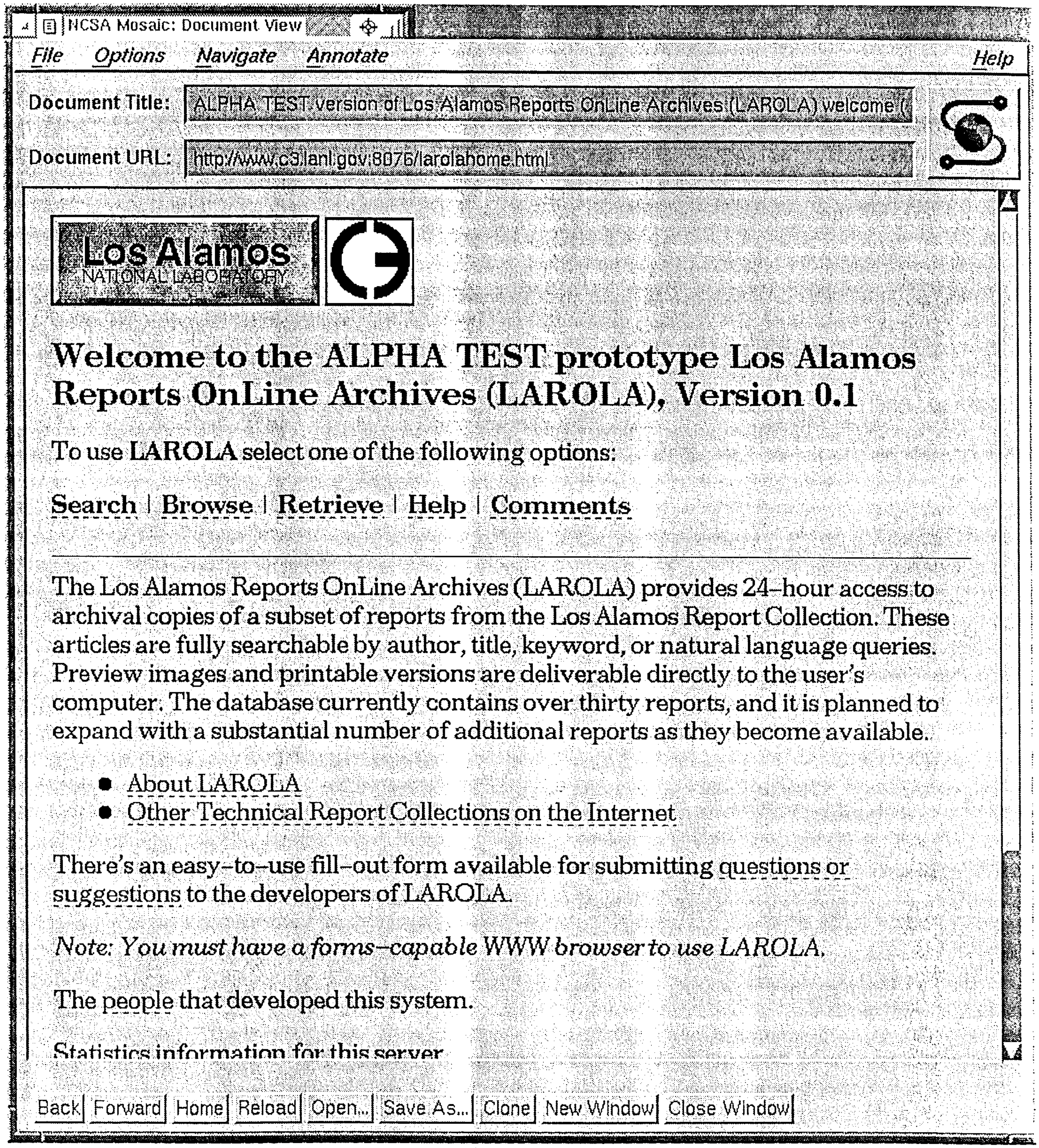
Printing is a difficult function for an image archive since the 300 dpi scanned images of a 200 page book occupy around 20 megabytes for compressed postscript. For this reason LAROLA provides a system for selecting the individual pages which the user wishes to print. It is envisioned that once users come to believe that the image will always be available on-line, they will greatly reduce the demand for printed versions.

CONCLUSION

We believe that LAROLA is the first deployed system in the world which combines the following features:

- 24 hour, global access
- Presents electronic images of archived paper documents
- Fielded searching on title, author, and year
- Searching on the full text in the body of the document
- Full Boolean searches with AND, OR, NOT, and ADJ
- Relevance Feedback to locate documents with similar topics
- Access to documents of arbitrary length even on small PCs
- Uses free browsing tools — no cost to user
- Allows printing of any combination of selected pages
- Help pages available for every screen

LAROLA is a universally available, working example showing how an archival system can seamlessly integrate the past – the substantial body of paper documents that form the foundations of scientific knowledge – with the future of online publishing.



NCSA Mosaic: Document View

File Options Navigate Annotate Help

Document Title: ALPHA TEST version of Los Alamos Reports OnLine Archives (LAROLA) welcome (

Document URL: http://www.c3.lanl.gov:8076/larolahome.html

Los Alamos NATIONAL LABORATORY

Welcome to the ALPHA TEST prototype Los Alamos Reports OnLine Archives (LAROLA), Version 0.1

To use **LAROLA** select one of the following options:

Search | **Browse** | **Retrieve** | **Help** | **Comments**

The Los Alamos Reports OnLine Archives (LAROLA) provides 24-hour access to archival copies of a subset of reports from the Los Alamos Report Collection. These articles are fully searchable by author, title, keyword, or natural language queries. Preview images and printable versions are deliverable directly to the user's computer. The database currently contains over thirty reports, and it is planned to expand with a substantial number of additional reports as they become available.

- About LAROLA
- Other Technical Report Collections on the Internet

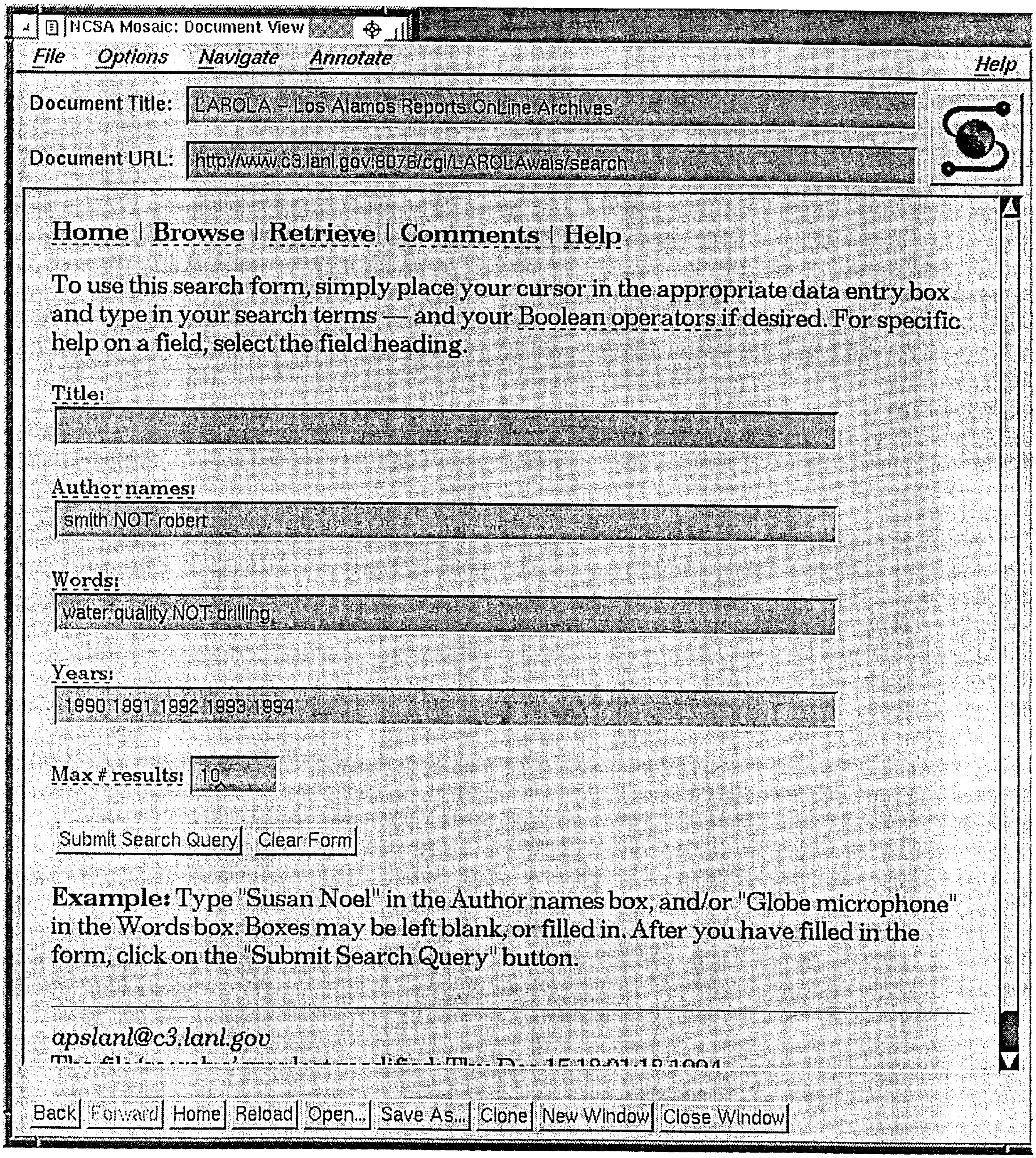
There's an easy-to-use fill-out form available for submitting questions or suggestions to the developers of LAROLA.

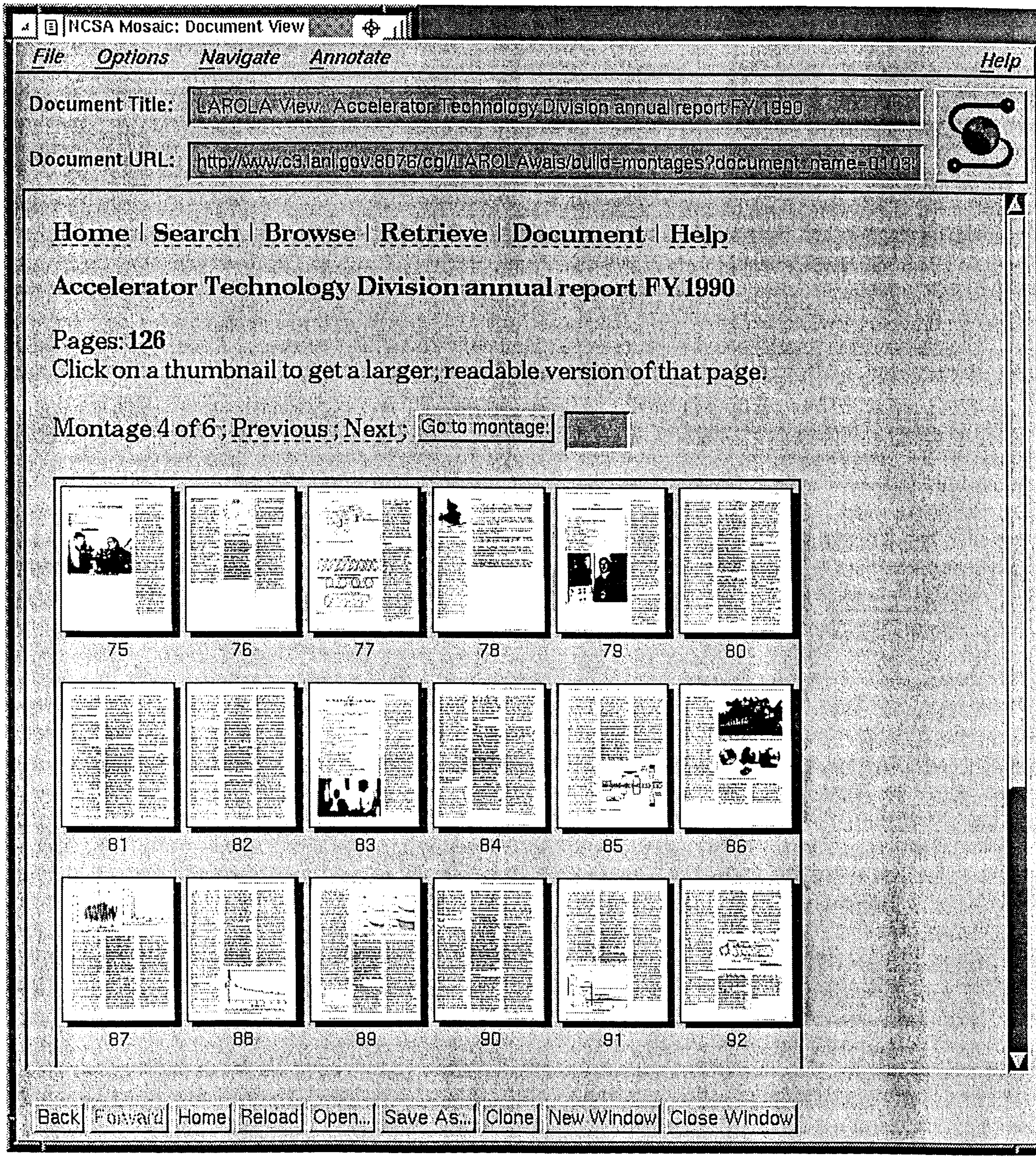
Note: You must have a forms-capable WWW browser to use LAROLA.

The people that developed this system.

Statistics information for this server

Back | Forward | Home | Reload | Open... | Save As... | Clone | New Window | Close Window


NCSA Mosaic: Document View
File Options Navigate Annotate Help
Document Title: LAROLA - Los Alamos Reports OnLine Archives
Document URL: http://www.c3.lanl.gov:8076/cgi/LAROLAwais/search
Home | Browse | Retrieve | Comments | Help
To use this search form, simply place your cursor in the appropriate data entry box and type in your search terms — and your Boolean operators if desired. For specific help on a field, select the field heading.
Title:
Author names:
smith NOT robert
Words:
water quality NOT drilling
Years:
1990 1991 1992 1993 1994
Max # results: 10
Submit Search Query
Clear Form
Example: Type "Susan Noel" in the Author names box, and/or "Globe microphone" in the Words box. Boxes may be left blank, or filled in. After you have filled in the form, click on the "Submit Search Query" button.
apslanl@c3.lanl.gov
Back Forward Home Reload Open... Save As... Clone New Window Close Window


NCSA Mosaic: Document View
File
Options
Navigate
Annotate
Help
Document Title:
LAROLA View: Accelerator Technology Division annual report FY 1990
Document URL:
http://www.c3.lanl.gov:8076/cgi/LAROLA/wais/build=montages?document_name=0103
Home | Search | Browse | Retrieve | Document | Help
Accelerator Technology Division annual report FY 1990
Pages: 126
Click on a thumbnail to get a larger, readable version of that page.
Montage 4 of 6 ; Previous ; Next ; Go to montage:
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
Back
Forward
Home
Reload
Open...
Save As...
Clone
New Window
Close Window

NCSA Mosaic: Document View

File Options Navigate Annotate Help

Document Title: LAROLA Inline Preview: Migration to a distributed system architecture at the Nationa

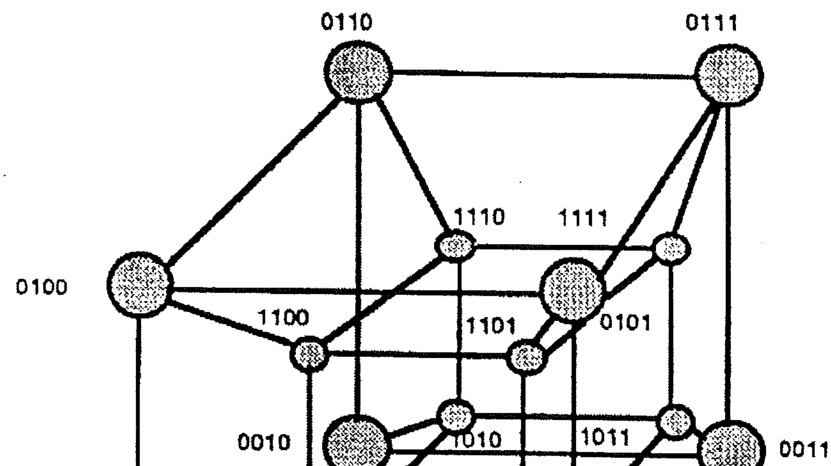
Document URL: http://www.c3.lanl.gov:8076/cgi/LAROLAwais/display-inline?document_name=010395

Home | Search | Browse | Retrieve | Document | Montage | Help

Migration to a distributed system architecture at the ...

Image 31 of 58. Previous | Next | Go to Image: View Text | Print

if necessary. It is not a requirement that these adaptors be in a single cabine close proximity. Problems may arise when considering the combined requi of the node and cluster backplanes.



Back Forward Home Reload Open... Save As... Clone New Window Close Window

Extending HTML Functionality with HyTime

Poster Summary

Lloyd Rutledge, John F. Buford, and John L. Rutledge
Distributed Multimedia Systems Laboratory
Computer Science Department
University of Massachusetts Lowell
One University Avenue
Lowell, MA 01854 USA
email: {lrutledg,buford,jrutledg}@cs.uml.edu

Introduction

Hypertext Markup Language (HTML) provides a document format for basic distributed hypermedia. However, it fails to meet the anticipated needs of the next wave of open and integrated hypermedia document usage. Standard Generalized Markup Language (SGML) and Hypermedia/Time-based Structuring Language (HyTime) meet many of these needs. The close relationship SGML has with both HTML and HyTime facilitates the incorporation of HyTime and additional SGML constructs into HTML processing. Such incorporation would help prevent the obsolescence of HTML documents as hypermedia environments become more open and integrated. It would also facilitate the incorporation of additional hypermedia functionality already defined by HyTime.

HTML is defined using an SGML document type definition (DTD). A DTD defines a set of allowable tags that can be used in structuring a document. A document instance is parsed with its DTD into the final SGML representation of the document. HyTime defines a set of patterns that can occur in this final document parse. These patterns represent the hypermedia structuring of the document.

Main Results

In this poster we describe four approaches for processing HTML documents within SGML and HyTime environments. Each approach uses different characteristics of how a document instance is parsed with its DTD. The advantages and disadvantages of each are weighed. The ordering of these approaches represents a progression from the alteration of legacy documents and document constructs to the processing of them in their original form.

An example is given for each approach. The examples all involve the functionality around the HTML anchor element type. The anchor represents a hyperlink between itself and another HTML-defined document object.

The complete revision approach involves rewriting the HTML DTD to use HyTime constructs. The advantage of this approach is that it allows a cleaner and more straightforward implementation of HyTime than the other approaches. The disadvantage is that legacy HTML documents will likely not be processable the resulting DTD.

The additional construct approach adds HyTime constructs to the HTML DTD without modifying any of its original contents. The advantage of this approach is that legacy HTML documents can be parsed with the new DTD. The disadvantage is that legacy HTML constructs will not parse into HyTime constructs.

The restructuring DTD approach modifies the HTML DTD so that legacy document instances parse with it into HyTime constructs. The advantage of this approach is that legacy documents do not have to be rewritten to be HyTime-conforming. The disadvantage is that only certain types of restructuring can be accomplished.

The overlaying document approach is the creation of a second, overlaying document that references portions of an HTML document. This overlaying document accesses objects in the HTML document and assigns HyTime-defined hypermedia characteristics to those objects. The document is small and can be applied to any HTML document without modification. The advantage of this approach is that legacy document instances parse with the legacy DTD into their original structures. The disadvantage is that more complex HyTime facilities are used than in the other approaches.

Conclusion

In this poster we show four approaches for implementing HyTime in HTML processing. We demonstrate that HyTime can be incorporated in legacy HTML documents without modification of those document instances.

WISKIT

WOMEN IN SCIENCE KIT†

Development of a multimedia software application

Laura Bright, W. John Burns, James Ford, Fillia Makedon
Charles Owen, Samuel Rebelsky, Nancy Toth, Qin Zhang

Dartmouth College
Fillia.Makedon@dartmouth.edu

Very few women and minorities know that computer science is a field of enormous opportunities and great excitement. In spite of fascinating new developments in the field, in terms of communication technology, multimedia, information infrastructure, parallel computing, the gap between the number of men and the number of women studying computer science in college is widening. From our experience, we have found that high school women often have a hard time relating to computer science as a future career, primarily because they do not know what it is about or because they do not know of any female computer scientist first hand, or because they cannot see themselves in that role. This is quite different from the case of, say, biology, physics, chemistry, all of which are older and more established fields, that also carry a certain romanticism with them as well as more concrete objects of study. Computer science, on the other hand, has just started being taught as a special topic in high schools, it is relatively new, fast-evolving and appears impersonal. For this reason, we feel it is very important, as teachers of computer science, to tell first-hand, what computer science is all about, and to do so with means that bring to life instances of scientists, students, university environments. The main objective of this kit will be to attract women and minorities to study computer science.

WISKIT was developed as an interactive multimedia information "kit" that explains computer science and related areas, such as computer engineering, to students who are about to enter college. The **WISKIT-CD** software is a Hypercard-based application that can be used by a variety of people such as, college and high school administrators, by students already in college (e.g., freshmen and sophomores), high school teachers and counselors who need an additional resource to counsel students, and even parents. **WISKIT-WWW** contains essentially the same information, but in on-line form on the World Wide Web[1]. Either version can also play an important role for those students already in college who wish to decide about a field of graduate study. They are a kaleidoscopic collection of what computer scientists do, who they are, and how they reached where they are.

WISKIT was designed to incorporate materials which give examples of many successful women in the field at different stages in their career, starting with undergraduate students. We want to provide cases which emphasize that computer science is a very suitable field for a woman who wishes to combine family and career since it allows for flexible hours of work from home. Due to the convergence of many fields (*e.g.*, biology and computer science), computer science provides opportunities for interdisciplinary work and applications, from library science, to working for the government, to commercial applications. In other words, the flexibility of job choices offered by computer science is something that is not as limited as, say, a laboratory scientist who is attached to the techniques, materials and methodologies of a particular laboratory.

WISKIT is a unique informational tool that, in our estimation, is greatly needed. Its main benefit is the inspiration it provides to women to become involved in computer science (or science in general, for that matter). A secondary benefit is that it has provided a prototype of how similar tools can be constructed for other fields or specializations, or companies. The organization and integration of multimedia documents is not an easy problem and we have added significantly to our expertise over the course of this project.

---

†This work has been supported by NECUSE (the New England Consortium for Undergraduate Science Education) and WISP (the Women in Science Program at Dartmouth College).

[1] http://www.cs.dartmouth.edu/~wiskit

Birkhäuser
Boston • Basel • Berlin

ISBN 0-8176-3846-6